Uma técnica de penalidade exacta baseada numa procura curvilínea em optimização não linear

Maria Teresa Torres Monteiro

Dissertação submetida à Universidade do Minho
para obtenção do grau de doutor
no Ramo da Engenharia de Produção e Sistemas,
Área de Métodos Numéricos e Estatísticos

Braga, Dezembro de 2000

Conteúdo

Resumo iv

Abstract vi

Agradecimentos vii

Dedicatória viii

1 Introdução 1
- 1.1 O problema . . . 1
- 1.2 Motivação . . . 2
- 1.3 Objectivos e plano de trabalhos . . . 4
- 1.4 Estrutura da tese . . . 5

2 Sobre o método de Newton modificado 8
- 2.1 Método de Newton . . . 8
- 2.2 Modificações do método de Newton . . . 10
- 2.3 Método de Newton modificado . . . 12
- 2.4 Estudo da convergência local . . . 14
- 2.5 Tópicos sobre a convergência global . . . 21
- 2.6 Comentários, síntese e motivação . . . 23

3 Técnicas de globalização 25
- 3.1 Introdução . . . 25
- 3.2 Procura unidimensional . . . 26
  - 3.2.1 Algumas propriedades da direcção de procura . . . 27
  - 3.2.2 Cálculo do comprimento do passo . . . 27
  - 3.2.3 Procura não monótona . . . 33
- 3.3 Regiões de confiança . . . 36

3.4 Procura unidimensional *versus* regiões de confiança 39
3.5 Comentários e motivação 39

4 Optimização não linear com restrições **41**
4.1 Introdução 41
4.2 Função mérito 43
4.3 Programação quadrática sequencial 44
4.4 Técnicas de penalidade 47
4.4.1 Função de penalidade sequencial l_2 47
4.4.2 Função de penalidade exacta 48
4.5 Função Lagrangeana aumentada 49
4.6 Método dos filtros 52
4.7 Síntese 54

5 Método pseudo-Newton modificado **55**
5.1 Estimação dos multiplicadores de Lagrange 56
5.2 Parâmetro de penalidade 59
5.3 Pseudo-derivadas 60
5.4 Sistema pseudo-Newton 63
5.5 Análise da convergência local 66
5.6 Direcção pseudo-Newton modificada 69
5.7 Estudo da convergência global 73
5.8 Síntese e comentários 90

6 Procura curvilínea **92**
6.1 O estado da arte 92
6.2 Vector correcção 95
6.3 Procura unidimensional 96
6.4 Estudo da convergência global 99
6.5 Síntese e comentários finais 105

7 Resolução inexacta de sistemas **106**
7.1 Motivação 106
7.2 Método dos gradientes conjugados 107
7.2.1 Convergência 109
7.2.2 Variantes do método de Lanczos 110
7.3 O método de Newton truncado 112
7.4 Precondição 117

7.5 Comentários e implementação efectuada 118

8 Experiências computacionais **120**
8.1 Ambiente de programação 120
8.2 Análise de resultados 120

9 Conclusões e investigação futura **133**
9.1 Conclusões 133
9.2 Futuras direcções de investigação 135

A Tabelas de Resultados **138**

B Teoremas e Lemas **161**

Bibliografia **166**

Resumo

O objectivo principal desta tese é a resolução de problemas de optimização não linear com restrições de igualdade, utilizando uma técnica de penalidade exacta que inclui uma abordagem curvilínea. A função objectivo do correspondente problema sem restrições é uma função Lagrangeana aumentada cujo ponto estacionário se obtém através dum método do tipo Newton. O algoritmo combina uma estratégia do tipo Newton truncado com um esquema de procura curvilínea.

O processo iterativo pseudo-Newton modificado baseia-se em pseudo-derivadas e tem um efeito regularizador ao adicionar um múltiplo da matriz identidade a um múltiplo da inversa da pseudo-Hessiana, enquanto a direcção pseudo-Newton não for suficientemente descendente. A escolha destes múltiplos tem influência no desempenho do algoritmo.

Para induzir convergência global, recorre-se a uma técnica de procura unidimensional que é baseada numa condição de Armijo modificada, numa escolha do comprimento do passo inicial de procura e num esquema de redução sucessiva desse passo.

A direcção pseudo-Newton modificada apresenta propriedades teóricas interessantes. Mostra-se ainda que a condição necessária e suficiente para uma convergência do tipo q-superlinear do método pseudo-Newton modificado, é a de que os deslocamentos gerados por este método convergem, em tamanho e direcção, para o passo de Newton, a partir dos mesmos pontos.

Com o propósito de acelerar a convergência, implementou-se um esquema de procura curvilínea, adicionando um vector correcção à direcção pseudo-Newton modificada de forma a encaminhar o processo iterativo para a região admissível. Também se prova que a correspondente sucessão de iterandos converge q-superlinearmente para um ponto estacionário.

A resolução dos sistemas lineares é feita de forma inexacta utilizando-se uma versão do método dos gradientes conjugados, de forma a reduzir a carga computacional do algoritmo final. O valor dos resíduos permite o controlo da eficiência do algoritmo.

Apresentam-se resultados computacionais para uma gama de problemas não lineares com restrições de igualdade retirados da base de problemas do CUTE. Estes resultados

indiciam que a estrutura algorítmica baseada no método pseudo-Newton modificado com procura curvilínea é, em termos práticos, eficiente e comparável a outros códigos conhecidos, como sejam o MATLAB e o NPSOL.

Abstract

In this thesis, an exact penalty technique that incorporates a curvilinear approach is proposed for solving equality constrained nonlinear optimization problems. The corresponding unconstrained problem objective function is an augmented Lagrangian function where the stationary point is obtained through a Newton type method. The algorithm includes a combination of a truncated Newton type strategy and a curvilinear search scheme.

The modified pseudo-Newton iterative process is based on pseudo-derivatives and has a regularization effect by adding a multiple of the identity matrix to a multiple of the inverse pseudo-Hessian, as long as the pseudo-Newton direction is not sufficiently descent. The multiples choice become a performance issue in the algorithm.

Global convergence had to be enforced by line search procedures based on an appropriate modified Armijo condition, initial step size choices and a backtracking scheme.

The modified pseudo-Newton steps have interesting theoretical properties. It is also shown that a necessary and sufficient condition for the q-superlinear convergence of the modified pseudo-Newton method is that the modified steps converge, in magnitude and direction, to the Newton steps from the same points.

In order to accelerate the convergence, a curvilinear search scheme is implemented by adding a correction step to the modified pseudo-Newton step to force the iterate to the feasible set. It is also shown that the corresponding sequence of iterates converges q-superlinearly to a stationary point.

Inexact linear systems resolutions with a conjugate gradient based method are introduced to reduce the computational burden involved in the final algorithm. The magnitude of the residuals provides control on the efficiency of the algorithm.

Computational results are included for a variety of equality nonlinear problems from the CUTE collection. Preliminary numerical testing indicates that the modified pseudo-Newton curvilinear algorithmic framework is efficient in practice and comparable to other known codes, namely MATLAB and NPSOL.

Agradecimentos

É muito difícil conseguir expressar o meu reconhecimento à Doutora Edite Fernandes, orientadora deste projecto. Desde o primeiro instante, a Doutora Edite esteve sempre presente, disponível e atenta ao desenvolvimento deste trabalho, fornecendo ideias, apresentando críticas e sugestões e corrigindo-o de forma expedita e eficaz. Gostaria ainda de salientar, o ambiente de amizade e de apoio, científico e emocional, em que se desenvolveram as nossas relações de trabalho. Apenas isto - foi um privilégio ter sido orientada por ela.

Para os colegas do subgrupo de Métodos Numéricos e Estatísticos vai também o meu agradecimento, pelo apoio e pelos momentos que juntos partilhámos ao longo deste período. No entanto, gostaria de agradecer muito em particular, ao colega Ismael Vaz, pela disponibilidade e boa vontade que sempre demonstrou às minhas solicitações. A sua ajuda, em termos de conhecimentos informáticos, foi crucial em várias etapas deste trabalho.

À Universidade do Minho o meu agradecimento por todas as facilidades concedidas para a realização deste trabalho, nomeadamente o período de equiparação a bolseira.

Gostaria ainda de referir que este trabalho foi suportado pelo Programa PRODEP, Concurso nº1/96, PRODEP II.

Para toda a minha família, o meu marido e em especial os meus pais, pelo apoio principalmente nas alturas de desânimo, e por me terem ajudado a constatar que afinal "havia luz no fundo do túnel".

Aos meus filhos, Alexandre e agora também o Francisco, a quem dedico este trabalho, pela estranha força mágica que em mim desencadeiam e que foi, sem dúvida, o estímulo para levar a bom termo este trabalho.

Por fim, algumas palavras para o céu, em discurso directo - "Mani, pelos pensamentos e reflexões que em muitos momentos mantenho contigo, aqui vai também a minha gratidão e saudade..."

A todos, muito obrigada.

Dedicatória

"A ciência realmente só tem alcançado tornar mais intensa e forte uma certeza:
- a velha certeza socrática da nossa irreparável ignorância. De cada vez sabemos mais que não sabemos nada" . - Eça de Queiroz

Aos meus filhos, Alexandre e Francisco

Capítulo 1

Introdução

Uma estratégia possível para a resolução de um problema de optimização não linear com restrições de igualdade baseia-se na técnica de penalidade exacta. Esta tese explora as características desta técnica de modo a produzir um algoritmo eficiente e razoavelmente robusto.

Neste primeiro capítulo, introduz-se o problema de optimização não linear com restrições de igualdade. De seguida, apresentam-se os motivos que ditaram a escolha da técnica de penalidade exacta, bem como os objectivos e o plano de trabalhos. Por fim, descreve-se a forma como esta tese está estruturada.

1.1 O problema

Os problemas de optimização são constituídos por três elementos básicos: a função objectivo, que se pretende minimizar ou maximizar, o conjunto de incógnitas ou variáveis que afectam o valor da função objectivo e o conjunto de restrições que permitem que as variáveis tomem certos valores, excluindo outros.

A função objectivo pode apresentar diferentes características. Pode ser linear, quadrática, pode ter a forma de uma soma de quadrados, ou até pode mesmo nem existir. As variáveis são essenciais pois se não existirem não se pode definir a função objectivo nem as restrições. As restrições podem existir ou não dando origem a problemas de optimização, respectivamente, com ou sem restrições. No entanto, a maioria dos problemas reais são caracterizados pela existência de restrições nas variáveis. O conjunto de pontos que verifica todas as restrições define a região admissível. Tal como a função objectivo, as funções de restrição podem apresentar várias formas.

A optimização pode ser vista em duas vertentes - a maximização e a minimização. Os

problemas de maximização podem ser postos em termos de problemas de minimização, pelo que o problema de optimização vai ser, a partir de agora, referenciado como de minimização.

Num problema de minimização, o objectivo é encontrar os valores das variáveis que minimizam a função objectivo, satisfazendo as restrições impostas. A formulação geral deste problema é a seguinte:

$$\begin{array}{lll} min & f(x) & \\ x \in \mathbb{R}^n & & \\ \text{sujeito a} & c_i(x) = 0, & i = 1, \ldots, m, \\ & \widehat{c}_i(x) \geq 0, & i = 1, \ldots, t, \\ & l_i \leq x_i \leq u_i, & i = 1, \ldots, n, \end{array} \tag{1.1}$$

em que $c_i(x) = 0$, $i = 1, \ldots, m$, são as restrições do tipo igualdade, $\widehat{c}_i(x) \geq 0$, $i = 1, \ldots, t$, as restrições de desigualdade e $l_i \leq x_i \leq u_i$, $i = 1, \ldots, n$, são as restrições de limites simples. As restrições do tipo desigualdade e de limites simples podem ser transformadas em igualdades, através da introdução de variáveis de folga, aumentando-se, no entanto, a dimensão do problema. O problema analisado nesta tese apresenta apenas restrições de igualdade e tem a seguinte forma:

$$\begin{array}{lll} min & f(x) & \\ x \in \mathbb{R}^n & & \\ \text{sujeito a} & c_i(x) = 0, & i = 1, \ldots, m. \end{array}$$

1.2 Motivação

Existem vários métodos para resolver este tipo de problema, sendo os mais conhecidos o método de programação quadrática sequencial, o método dos gradientes reduzidos, os métodos baseados nas técnicas de penalidade sequencial e exacta, e mais recentemente o método dos filtros. Dependendo das características do problema, nomeadamente a dimensão e estrutura, assim se escolhe o método mais adequado à sua resolução ([16], [82]).

Também já existem vários pacotes de *software* adaptados a cada tipo de problema. É muito importante salientar que nenhum *software* resolve todo o tipo de problemas de optimização não linear - o importante é encontrar o mais eficiente para a resolução de cada problema, o que não é uma tarefa fácil. Enumera-se de seguida uma lista dos pacotes mais conhecidos e respectiva utilização. A selecção efectuada não é portanto exaustiva e tem em conta a experiência já adquirida no passado: CONOPT utilizado

para programação não linear genérica com restrições não lineares esparsas, DONLP2 para minimização de funções não lineares suaves com restrições suaves, GRG2 utilizado na programação não linear, HARWELL para programação linear e não linear, equações não lineares e ajuste de dados, IMSL para a optimização com e sem restrições e equações não lineares, LANCELOT utilizado para problemas de grandes dimensões, LOQO utilizado na programação linear, optimização não linear com e sem restrições, MATLAB (*toolbox* de optimização) para optimização não linear, MINOS para programação linear e optimização não linear com e sem restrições, NAG (Linguagem C ou Fortran) utilizada na programação quadrática e não linear e NPSOL para minimização de funções não lineares suaves. Um conjunto mais vasto de *software* de optimização pode ser consultado no livro de Moré e Wright [84].

Hoje em dia existem colecções de problemas teste de optimização disponíveis em bases de dados, algumas de livre acesso, de forma a permitir que os vários investigadores testem a robustez e eficiência dos seus algoritmos. O CUTE ([4]) (do inglês *Constrained and Unconstrained Testing Environment*) para além de possuir uma base de dados de livre acesso com inúmeros problemas, faculta ainda ferramentas essenciais para a sua resolução, como por exemplo, o vector gradiente e a matriz Hessiana da função objectivo. Outro ambiente de programação muito conhecido é o AMPL que é um *software* comercial, sendo possível obter uma versão limitada, *Student Edition*, a partir da sua página de endereço *http://www.ampl.com*. O facto de, apenas ser de livre acesso a versão limitada, torna-o preterido em favor do CUTE.

Da lista de pacotes mencionada, dois dos mais referenciados na literatura científica e tecnológica são o MATLAB ([74]) e o NPSOL ([51]). Numa fase preliminar do projecto optou-se por fazer uma selecção de alguns problemas do CUTE e tentar resolvê-los pelos dois códigos. Os resultados obtidos ([79]), como se esperava, mostraram que alguns problemas eram resolvidos apenas por um dos códigos e outros não eram resolvidos por nenhum dos dois pacotes. O facto de o MATLAB e o NPSOL serem baseados em algoritmos que assentam na programação quadrática sequencial, serviu de estímulo ao desenvolvimento de um algoritmo baseado noutra técnica, nomeadamente a técnica de penalidade exacta, que resolvesse os problemas de optimização que estes códigos comerciais não conseguiam, ou que melhorasse o comportamento dos já resolvidos. Uma técnica de penalidade transforma o problema original com restrições num outro sem restrições de certa forma relacionado com o primeiro. Este novo problema pode ter como função objectivo uma função Lagrangeana aumentada, cujo termo de penalidade é do tipo l_2.

Segue-se uma breve descrição dos dois pacotes. O NPSOL é um conjunto de subrotinas em Fortran 77 para resolução de problemas não lineares. Usa um algoritmo de programação quadrática sequencial, em que a direcção de procura é a solução do subproblema quadrático. A matriz Hessiana de cada subproblema quadrático é uma aproximação defi-

nida positiva, do tipo Quasi-Newton, à matriz Hessiana da função Lagrangeana. O passo de procura em cada iteração obriga a uma redução significativa de uma função mérito do tipo Lagrangeana aumentada. Cada subproblema quadrático é resolvido por um pacote de programação quadrática cujas propriedades melhoram a eficiência do algoritmo de programação quadrática sequencial. O MATLAB é uma linguagem para computação técnica e uma família de *toolboxes*. Uma das *toolboxes* é a de optimização que trata da minimização/maximização de funções não lineares. Esta usa o método de programação quadrática sequencial e a estimação da matriz Hessiana da função Lagrangeana é actualizada em cada iteração usando a fórmula Quasi-Newton BFGS. A procura unidimensional baseia-se numa função mérito.

1.3 Objectivos e plano de trabalhos

Tendo como base uma técnica de penalidade exacta, a primeira fase do projecto começou por um estudo do problema de optimização não linear sem restrições. No cálculo do ponto estacionário da sua função objectivo, optou-se pelo desenvolvimento de um método do tipo Newton uma vez que era possível ter acesso às derivadas através do CUTE. À semelhança de inúmeros trabalhos desenvolvidos na área, o novo método pretendia resolver algumas das limitações que surgem durante a implementação do método de Newton. Neste método, quando existe singularidade da matriz Hessiana, a direcção de procura não está definida e, quando a matriz é indefinida, a direcção resultante não é necessariamente descendente. A ideia básica subjacente ao método de Newton modificado, consiste na definição de uma direcção de procura baseada num esquema de combinação linear convexa da direcção de descida máxima e da direcção Newton, resultando em direcções bem definidas e descendentes.

Passou-se ao objectivo central deste trabalho que consistia no desenvolvimento de um algoritmo razoavelmente robusto, eficiente e pouco exigente em termos computacionais, assente na filosofia das técnicas de penalidade exacta para a resolução do problema de optimização não linear com restrições do tipo igualdade. Para estimação do vector dos multiplicadores de Lagrange, usou-se uma função de primeira ordem calculada a partir das condições de optimalidade do problema original. No cálculo do ponto estacionário da função Lagrangeana aumentada, numa perspectiva de método primal, recorreu-se ao método de Newton. No entanto, a complexidade das fórmulas do vector gradiente e da matriz Hessiana conduziu a um procedimento de simplificação das mesmas. Assim, foram definidos o vector pseudo-gradiente e a matriz pseudo-Hessiana, para aproximar respectivamente o gradiente e a Hessiana da função Lagrangena aumentada no sistema Newton, dando origem ao sistema designado pseudo-Newton. Como o problema foi transformado,

através da técnica de penalidade, num problema sem restrições, a modificação do método de Newton efectuada na fase inicial do trabalho, foi estendida ao sistema pseudo-Newton. Desta forma, a direcção resultante do sistema pseudo-Newton foi, quando necessário, modificada de maneira a tornar-se descendente, para garantir que o processo iterativo conduza a um mínimo. O método resultante foi designado por método pseudo-Newton modificado. Como nem sempre as condições dos problemas práticos a resolver são ideais, foi introduzida uma estratégia de globalização, do tipo procura unidimensional, tendo como base a condição de Armijo, para garantir a convergência do método.

Inerente ao desenvolvimento do algoritmo surgiu uma nova tarefa relacionada com a análise de convergência. O esforço dispendido nesta fase crucial do trabalho, conduziu aos seguintes resultados. Foi possível estabelecer certas condições que garantem que o método converge, numa pequena vizinhança do ponto estacionário, e que para certos problemas específicos conduzem a uma convergência q-quadrática. Foram enunciadas algumas propriedades interessantes da direcção pseudo-Newton modificada, isto é, foi possível provar, sob certas condições, que a direcção pseudo-Newton modificada é suficientemente descendente, é do tipo gradiente, é limitada em norma e origina uma redução significativa no valor da função. Provou-se que a convergência para um ponto estacionário é q-superlinear.

No sentido de melhorar o desempenho do algoritmo, experimentou-se adicionar à direcção de procura, um vector correcção, sempre que o iterando não pertencesse à região admissível. A associação destas duas componentes, a direcção pseudo-Newton modificada e o vector correcção, originou um esquema de procura que foi designado por procura curvilínea. Também para o algoritmo que define este processo iterativo, foi feita uma análise de convergência. Concluiu-se que a sucessão de aproximações gerada pela procura curvilínea converge para um ponto estacionário q-superlinearmente.

Na resolução de sistemas lineares, houve a preocupação de apresentar estratégias com vista a diminuir o tempo de CPU e o espaço de memória. Foi implementado um método de resolução inexacta baseado em subespaços de Krylov, do qual se destaca o método dos gradientes conjugados.

Para validar os estudos teóricos efectuados, seleccionaram-se os mesmos problemas do CUTE que já tinham sido testados com o MATLAB e NPSOL. As diversas variantes que foram surgindo nas diferentes fases deste projecto foram testadas com os problemas seleccionados. Os resultados obtidos foram comparados e foi feito um estudo de avaliação dos códigos desenvolvidos em termos de robustez e eficiência.

1.4 Estrutura da tese

Esta tese está estruturada em nove capítulos e possui também dois apêndices.

No Capítulo 2 estuda-se a resolução do problema de optimização não linear sem restrições, com a introdução de modificações ao método de Newton clássico e que origina o método de Newton modificado. Faz-se uma retrospectiva das várias abordagens propostas por outros autores que, tal como o método de Newton modificado, têm como objectivo colmatar as limitações que surgem durante a implementação do método de Newton. Estuda-se também a convergência local do método e apresentam-se alguns resultados sobre a convergência global.

No Capítulo 3 apresentam-se soluções para a globalização dos algoritmos. Dá-se uma ênfase especial à técnica de procura unidimensional, por ter sido esta a seleccionada para este trabalho. No entanto, faz-se uma introdução ao outro tipo de técnica baseada em regiões de confiança.

No Capítulo 4 é introduzido o problema de optimização não linear com restrições, e são apresentadas várias estratégias para a sua resolução, a saber, a programação quadrática sequencial, as técnicas de penalidade sequencial e exacta e, mais recentemente, a baseada no conceito de filtros.

O quinto capítulo surge como crucial nesta tese. Estuda-se o método de Newton modificado, agora aplicado ao problema com restrições do tipo igualdade. Utilizou-se uma estratégia de penalidade exacta e são apresentadas a função mérito, utilizada nos contextos de globalização do algoritmo e de cálculo de aproximações à solução, a função de estimação do vector dos multiplicadores de Lagrange e um procedimento adaptativo para o parâmetro de penalidade. A complexidade das fórmulas das derivadas envolvidas, levou a que fossem introduzidas algumas simplificações no gradiente e na Hessiana da função mérito. São então introduzidas as pseudo-derivadas de primeira e segunda ordens e identificado o sistema pseudo-Newton. O método de Newton modificado, apresentado no Capítulo 2, quando aplicado ao sistema pseudo-Newton dá origem ao método pseudo-Newton modificado. É feita a análise da convergência local e são enunciadas algumas propriedades da direcção resultante. A introdução da estratégia de globalização originou uma convergência global do método q-superlinear.

No Capítulo 6 introduz-se um mecanismo de aceleração do processo iterativo, adicionando-se um vector correcção à direcção pseudo-Newton modificada, que tem como objectivo encaminhar o processo iterativo para a região admissível. A este novo esquema chamou-se procura curvilínea e sobre ele efectuou-se um estudo de convergência.

O Capítulo 7 foi destinado à resolução inexacta do sistema linear obtido pela aplicação do método pseudo-Newton modificado. São identificadas e analisadas algumas estratégias de resolução inexacta de sistemas, nomeadamente no que diz respeito aos métodos iterativos de Krylov de que o método dos gradientes conjugados é o exemplo mais conhecido. A utilização de um critério de terminação do processo iterativo que garanta o cálculo de uma aproximação adequada à direcção Newton, origina o algoritmo do método pseudo-Newton

modificado e truncado.

A análise das experiências computacionais realizadas ao longo deste projecto, ditaram a construção de tabelas com estatísticas e valores acumulados, e que são apresentadas no Capítulo 8, das quais foi possível mais facilmente tirar conclusões sobre o comportamento dos diversos algoritmos. As tabelas exaustivas encontram-se no Apêndice A. No Apêndice B são enunciados os lemas e teoremas mais utilizados nesta tese.

Por fim, no Capítulo 9 são apresentadas as conclusões finais deste trabalho e feitas sugestões para trabalho futuro.

Capítulo 2

Sobre o método de Newton modificado

Neste capítulo vai apresentar-se um método, do tipo Newton, para a resolução de problemas não lineares sem restrições nas variáveis. O problema a resolver tem a forma geral

$$\begin{array}{l} min \\ x \in IR^n \end{array} \quad f(x) \qquad \text{(P)}$$

em que $f : IR^n \longrightarrow IR$ é duas vezes continuamente diferenciável num conjunto aberto $D \subset IR^n$.

Começa-se por descrever o método de Newton clássico evidenciando a sua importância no contexto da optimização não linear, realçando no entanto as suas limitações a nível de implementação, que constituem por si só, a motivação para a modificação apresentada neste trabalho. Faz-se uma síntese das sugestões que têm surgido na literatura para modificar a matriz Hessiana. A seguir descreve-se o método de Newton modificado que é o assunto central deste capítulo. Estabelecem-se condições para uma convergência local quadrática do método de Newton modificado e no final faz-se um resumo sobre um estudo já realizado relativo à convergência global e algumas propriedades do mesmo.

2.1 Método de Newton

Uma estratégia possível e muito utilizada para resolver o problema (P) é o método de Newton, cuja equação iterativa é

$$x_{k+1} = x_k + d_{N_k} \qquad k = 0, 1, 2, \ldots ,$$

em que

$$\nabla^2 f(x_k) d_{N_k} = -\nabla f(x_k)$$

é a direcção Newton na iteração k, $\nabla f(x_k)$ e $\nabla^2 f(x_k)$ são respectivamente o vector gradiente e a matriz Hessiana de $f(x)$ em x_k.

Em [27] Dennis e Schnabel salientam que, no contexto da optimização, o método de Newton se apresenta como um método iterativo excepcionalmente eficiente, para qualquer problema em qualquer área. De facto, a sua simplicidade, a razão de convergência local q-quadrática e o facto de ser invariante quando se aplica algum esquema de escalonamento das variáveis independentes, são razões suficientes para fundamentar a sua importância.

Se x_0 estiver suficientemente próximo de x^* e x^* for um ponto estacionário de $f(x)$, ou seja, $\nabla f(x^*) = 0$, com $\nabla^2 f(x^*)$ não singular, então o método de Newton converge q-quadraticamente para x^*. O teorema seguinte (Teorema 5.2.1 em [28, pp 90]) descreve as condições de convergência do método de Newton.

Teorema 2.1 (Método de Newton) *Seja $\nabla f(x) : \mathbb{R}^n \to \mathbb{R}^n$ continuamente diferenciável no conjunto aberto D ($D \subset \mathbb{R}^n$). Assuma-se que existe $x^* \in \mathbb{R}^n$ e $r, \overline{\beta} > 0$, tal que $B_{x^*}(r) \subset D$, $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)^{-1}$ existe e $\left\|\nabla^2 f(x^*)^{-1}\right\| \leq \overline{\beta}$, e $\nabla^2 f(x)$ é contínua à Lipschitz em $B_{x^*}(r)$ com constante $\overline{K}$. Então existe $\varepsilon > 0$ tal que para todo $x_0 \in B_{x^*}(\varepsilon)$ a sucessão $x_1, x_2, \ldots$ gerada por*

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k), \qquad k = 0, 1, \ldots$$

é bem definida, converge para x^* e obedece a

$$\|x_{k+1} - x^*\| \leq \overline{\beta}\,\overline{K}\,\|x_k - x^*\|^2, \qquad k = 0, 1, \ldots$$

O método de Newton, se convergir, converge para um ponto estacionário. São necessárias estratégias auxiliares para que ele convirja para um mínimo. Será um mínimo local de f se e só se $\nabla^2 f(x^*)$ for uma matriz definida positiva (Corolário 4.3.3 de [28]).

Todavia, o método apresenta algum dispêndio de memória e de tempo, para o cálculo e armazenamento da matriz $\nabla^2 f(x_k)$ bem como para a resolução do sistema, se a dimensão do problema for considerada razoável. Na prática, nem sempre é possível dispor de uma aproximação inicial x_0 suficientemente próxima de x^* e ainda $\nabla^2 f(x_k)$ pode não ser definida positiva. Há situações onde $\nabla^2 f(x_k)^{-1}$ pode mesmo não existir e portanto d_{N_k} não estar definida. Embora em teoria ele seja visto como "ideal", estas características justificam que, na prática, ele não seja usado na sua forma clássica. Surgem então as modificações do método de Newton.

2.2 Modificações do método de Newton

Têm surgido, ao longo dos tempos, várias modificações ao método de Newton com o objectivo de responder, de um modo eficiente, às suas limitações. Faz-se de seguida, uma descrição sucinta das modificações mais conhecidas que se resumem a técnicas de modificação da matriz Hessiana, e, no âmbito dos mínimos quadrados faz-se referência aos métodos Gauss-Newton e de Levenberg-Marquardt.

O objectivo das alterações introduzidas na matriz Hessiana, para o cálculo da direcção de procura, consiste em transformá-la numa matriz definida positiva. Para que a informação de segunda ordem seja preservada, a modificação deve ser a mais ligeira possível e, em termos computacionais, não deve envolver custos adicionais significativos.

Uma estratégia baseada na decomposição espectral, consiste na substituição dos valores próprios negativos, por valores pequenos e positivos ligeiramente superiores à precisão da máquina. Outra sugestão, é a alteração do sinal dos valores próprios negativos. Estas modificações, consideradas teoricamente adequadas, para além de envolverem o cálculo dos valores próprios da matriz original, implicam ainda a composição da nova matriz com alguns dos valores e respectivos vectores próprios alterados, o que em termos de cálculos e esforço computacional se torna muito pesado ([87]).

Outra técnica baseia-se na adição dum múltiplo da matriz identidade à matriz Hessiana original, que deverá ser maior do que o valor absoluto do valor próprio mais negativo. Na prática, nem sempre se dispõe duma estimativa para esse valor, recorrendo-se ao facto da norma de Frobenius duma matriz ser limite superior do seu valor próprio de maior módulo. Desta forma, a matriz a adicionar à matriz Hessiana será diagonal, com todos os elementos iguais à norma de Frobenius. Este facto origina um deslocamento, no sentido positivo, de todos os valores próprios da matriz Hessiana original. Esse deslocamento pode ser desnecessariamente grande, desequilibrando a direcção Newton modificada ([87]). Em [28] é sugerida uma outra modificação a $\nabla^2 f(x_k)$,$\overline{\nabla^2 f}(x_k) = \nabla^2 f(x_k) + \lambda_k I$, em que λ_k é ligeiramente superior ao módulo do valor próprio mais negativo de $\nabla^2 f(x_k)$, estimado pelo teorema dos círculos de Gerschgorin. A factorização Cholesky é então aplicada a $\overline{\nabla^2 f}(x_k)$.

Quando $\nabla^2 f(x_k)$ não é definida positiva, é possível tornar a matriz dos coeficientes do sistema Newton definida positiva, através da adição de uma matriz diagonal, $\nabla^2 f(x_k) + D_k$. A factorização de Cholesky modificada, apresentada em [53], calcula assim a matriz diagonal, D_k, a partir dos elementos de $\nabla^2 f(x_k)$, com elementos não negativos, de tal forma que $\overline{\nabla^2 f}(x_k) = \nabla^2 f(x_k) + D_k = L_k^T L_k$ seja definida positiva.

Outra estratégia para modificar uma matriz indefinida recorre a um procedimento baseado numa factorização simétrica indefinida conhecida por factorização LBL^T de Bunch

e Parlett em que $P^T \nabla^2 f\, P = LBL^T$, com P uma matriz de permutação, B uma matriz diagonal com blocos ($B = diag(B_1, B_2, \ldots, B_j)$, $j = 1, \ldots, n$) e L uma matriz triangular inferior com uns sobre a diagonal. Cada B_j é uma matriz de 1×1 ou 2×2 não singular (ver detalhes em [87, pp 151]).

Em [108] é proposto um esquema de segurança que identifica as dificuldades surgidas durante a implementação do método de Newton, que se resumem a

i) singularidade de $\nabla^2 f(x_k)$,

ii) ortogonalidade de d_{N_k} relativamente a $\nabla f(x_k)$,

iii) não descendência de d_{N_k} em relação a $f(x)$ em x_k

e sugere alternativas.

Esta estratégia ([108]) aceita a direcção de descida máxima ($-\nabla f(x_k)$) quando o método de Newton falha devido a i) e ii) e substitui d_{N_k} por $-d_{N_k}$ quando iii) ocorre. O trabalho de Grippo, Lampariello e Lucidi [61] também sugere este procedimento.

Uma outra alternativa que tem originado grande sucesso introduz nas equações do sistema Newton uma modificação de tal forma que a direcção resultante tenha uma contribuição ao longo da direcção de descida máxima. Em cada iteração, é resolvido o problema

$$\begin{array}{ll} min & \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \\ d \in \mathbb{R}^n & \text{sujeito a } \|d\| \leq \Delta_k, \end{array}$$

em ordem a d_k, que é equivalente a (Lema 6.4.1 de [28, pp 131])

$$d_k = -\left(\nabla^2 f(x_k) + \mu_k I\right)^{-1} \nabla f(x_k)$$

para algum $\mu_k \geq 0$ (veja-se também secção 5.2 em [43]). Esta formulação identifica a técnica de regiões de confiança. A convergência global para um ponto estacionário pode facilmente ser estabelecida (veja-se em [88]). Para os outros métodos do tipo Newton referidos atrás, a convergência global pode ser estabelecida num contexto de procura unidimensional.

No contexto dos problemas de mínimos quadrados também existem estratégias de modificação da matriz Hessiana. No método Gauss-Newton, excluem-se os termos de segunda ordem da matriz Hessiana da função objectivo, trazendo vantagens significativas. Para além de se evitar o cálculo das segundas derivadas dos resíduos, para problemas de resíduos nulos este método tem uma convergência local quadrática ([28]). Existe ainda equivalência entre as equações do método Gauss-Newton e as equações normais dos problemas de

mínimos quadrados lineares. O método Gauss-Newton resolve problemas de mínimos quadrados lineares numa única iteração. No entanto, apresenta algumas desvantagens. Para problemas de resíduos grandes a convergência é lenta, não é bem definido se a matriz do Jacobiano dos resíduos não tiver característica completa e não tem necessariamente convergência global.

No que diz respeito ao método de Levenberg-Marquardt (veja-se por exemplo [28], [81] e [108]), embora tenha sido originalmente aplicado no domínio dos problemas de mínimos quadrados, é geralmente considerado como o progenitor da abordagem de regiões de confiança para a optimização sem restrições. À matriz do sistema Gauss-Newton é adicionado um múltiplo da matriz identidade por forma a afastá-la da singularidade.

2.3 Método de Newton modificado

Com o propósito de resolver algumas das limitações do método de Newton e à semelhança dos trabalhos já referidos, apresenta-se um método alternativo que será designado ao longo deste trabalho por método de Newton modificado. O método iterativo proposto é um método descendente no sentido em que gera direcções descendentes.

Definição 2.1 *Seja $f(x)$ uma função continuamente diferenciável em $D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$. Uma direcção d_k diz-se descendente em relação a f em x_k,se*

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

Definição 2.2 *Seja $f(x)$ uma função continuamente diferenciável em $D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$. Uma direcção d_k diz-se suficientemente descendente se existe um $\eta_1 > 0$ tal que*

$$\nabla f(x_k)^T d_k \leq -\eta_1 \|\nabla f(x_k)\| \|d_k\|.$$

A filosofia deste método é semelhante à utilizada por Wolfe [108] para o método de segurança de Newton. A ideia básica do método de Newton modificado consiste na construção duma nova direcção de procura, d_{NM_k}, que é uma combinação linear convexa da direcção Newton com a de descida máxima. Quando a matriz Hessiana é singular, d_{N_k} não está definida, e a combinação linear deve originar uma direcção descendente. Seja esta direcção a de descida máxima, $-\nabla f(x_k)$. Assim, para $w(x_k) = 1$, em

$$d_{NM_k} = (1 - w(x_k))d_{N_k} - w(x_k)\nabla f(x_k),$$

tem-se $d_{NM_k} = -\nabla f(x_k)$. Quando d_{N_k} não é uma direcção suficientemente descendente em relação a f em x_k, a função peso $w(x_k)$ deve ser tal que a nova direcção, d_{NM_k}, satisfaça $\nabla f(x_k)^T d_{NM_k} < 0$.

Assim, de

$$-\nabla f(x_k)^T \left[(1-w(x_k))\nabla^2 f(x_k)^{-1} + w(x_k)I\right] \nabla f(x_k) < 0$$

obtém-se

$$w(x_k) > \frac{\nabla f(x_k)^T d_{N_k}}{\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}}$$

(veja-se em [39]).

O novo processo iterativo é definido por

$$x_{k+1} = x_k + d_{NM_k}, \qquad k = 0, 1, 2, \ldots, \tag{2.1}$$

em que

$$d_{NM_k} = -\left[(1-w(x_k))\nabla^2 f(x_k)^{-1} + w(x_k)I\right] \nabla f(x_k) \tag{2.2}$$

é a direcção de procura do método de Newton modificado e $w(x_k)$ é uma função peso, $w : \mathbb{R}^n \longrightarrow \mathbb{R}$, calculada por

$$w(x_k) = \frac{\nabla f(x_k)^T d_{N_k}}{\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}} + \gamma, \tag{2.3}$$

com $\gamma > 0$ tal que $0 \leq w(x_k) \leq 1$. Esta condição implica que

$$\frac{\nabla f(x_k)^T d_{N_k} + \gamma\left(\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}\right)}{\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}} - 1 \leq 0.$$

Com

$$\begin{aligned} \|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k} \quad &> \quad 0 \quad \text{para} \quad \nabla f(x_k)^T d_{N_k} \geq 0 \\ &\Longrightarrow \quad \gamma\left(\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}\right) - \|\nabla f(x_k)\|_2^2 \leq 0 \end{aligned}$$

vem

$$0 < \gamma \leq \frac{\|\nabla f(x_k)\|_2^2}{\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}}.$$

Assim, para que $0 \leq w(x_k) \leq 1$ deve ter-se

$$w(x_k) = \frac{\nabla f(x_k)^T d_{N_k} + \varepsilon \|\nabla f(x_k)\|_2^2}{\|\nabla f(x_k)\|_2^2 + \nabla f(x_k)^T d_{N_k}} \text{ com } 0 < \varepsilon \leq 1. \tag{2.4}$$

Esta função peso define uma direcção descendente em relação a $f(x)$ em x_k.

O procedimento que usa as equações (2.1), (2.2) e (2.4) gera o algoritmo de Newton modificado. Quando a direcção Newton é suficientemente descendente para $f(x)$ em x_k, o algoritmo atribui a $w(x_k)$ o valor 0 e d_{N_k} é aceite, isto é, $d_k^{MN} = d_{N_k}$. Esta estratégia é descrita no seguinte algoritmo:

Algoritmo 2.1 (Método de Newton modificado)

Dados η_1, ε, x_0;
Enquanto (`Critério de paragem = falso`) ***Fazer***
  Se $(\nabla^2 f(x_k)$ *é singular)* ***Então*** $w(x_k) = 1$;
  Senão {*calcular* d_{N_k} *- solução de* $\nabla^2 f(x_k) d_{N_k} = -\nabla f(x_k)$ } ;
   Se $(\nabla f(x_k)^T d_{N_k} \leq -\eta_1 \|\nabla f(x_k)\| \|d_{N_k}\|)$ ***Então*** $w(x_k) = 0$;
    Senão {*calcular* $w(x_k)$ *usando (2.4)* };
   FimSe;
  FimSe;
{*calcular* $x_{k+1} = x_k - w(x_k)\nabla f(x_k) + (1 - w(x_k)) d_{N_k}$};
FimEnquanto

Este algoritmo tem a seguinte interpretação em termos de um modelo quadrático da função f em x_k: a direcção d é o minimizante da aproximação

$$m_q(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) \left[w(x_k) \nabla^2 f(x_k) + (1 - w(x_k)) I \right]^{-1} d.$$

2.4 Estudo da convergência local

Nesta secção vai ser feito o estudo da razão de convergência local da sucessão $\{x_k\}$ produzida pelo processo iterativo de Newton modificado (2.1), (2.2) e (2.4).

Seja $\|.\|$ a norma Euclideana em $\mathbb{R}^n$ ou o operador norma induzido no espaço de matrizes reais de ordem n. $B_r(x^*)$ representa a bola aberta de raio r e centro x^*, isto é,

$$B_r(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\| < r\}$$

para $r > 0$. Assuma-se que $B_r(x^*) \subset D$, em que D é um conjunto aberto.

A solução do problema (P) é representada por x^*. Assuma-se ainda que quando $\{x_k\}$ converge para x^*, se verifica $x_k \neq x^*$ para valores de k suficientemente grandes.

Considerem-se as seguintes hipóteses:

Hipótese 2.1 *A função $f(x)$ é duas vezes continuamente diferenciável num conjunto aberto $D \subset \mathbb{R}^n$.*

Hipótese 2.2 *x^* é a solução de (P) e $\nabla f(x^*) = 0$.*

Hipótese 2.3 *Existe uma constante $\overline{\beta} > 0$ tal que $\nabla^2 f(x^*)$ é não singular e $\|\nabla^2 f(x^*)^{-1}\| \leq \overline{\beta}$.*

Hipótese 2.4 *$\nabla^2 f(x)$ é contínua à Lipschitz em x^* com constante $\overline{K}$ $(0 < \overline{K} < \infty)$*

$$\left\|\nabla^2 f(x_k) - \nabla^2 f(x^*)\right\| \leq \overline{K}\,\|x_k - x^*\|$$

com $x_k \in B_r(x^*)$.

Considere-se o teorema fundamental do cálculo ([66, pp 65]):

Teorema 2.2 *Seja $\nabla f(x)$ diferenciável num conjunto aberto $D \subset \mathbb{R}^n$ e $x^* \in D$. Então para qualquer $x \in D$ suficientemente próximo de x^* e $0 \leq t \leq 1$*

$$\nabla f(x) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + t(x - x^*))(x - x^*)\,dt.$$

O lema seguinte é fundamental para o estudo da convergência.

Lema 2.1 *(Lema 4.3.1 de [66, pp 69]) Considerem-se as Hipóteses 2.1, 2.2, 2.3 e 2.4. Então existe uma constante $\delta > 0$ tal que para todo o $x \in B_\delta(x^*)$*

$$\begin{aligned} i) \quad \|\nabla^2 f(x)\| &\leq 2\|\nabla^2 f(x^*)\|; \\ ii) \quad \|\nabla^2 f(x)^{-1}\| &\leq 2\|\nabla^2 f(x^*)^{-1}\|; \\ iii) \quad \|\nabla f(x)\| &\leq 2\|\nabla^2 f(x^*)\|\|x - x^*\|. \end{aligned}$$

Prova. Para provar $i)$, considere-se δ suficientemente pequeno tal que $B_\delta(x^*) \subset D$. Para todo $x \in B_\delta(x^*)$, a Hipótese 2.4 e a desigualdade triangular implicam

$$\|\nabla^2 f(x)\| - \|\nabla^2 f(x^*)\| \leq \|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq \overline{K}\|x - x^*\|$$

obtendo-se

$$\|\nabla^2 f(x)\| \leq \|\nabla^2 f(x^*)\| + \overline{K}\delta$$

e reduzindo δ se necessário de tal forma que $\overline{K}\delta \leq \|\nabla^2 f(x^*)\|$ obtém-se $\|\nabla^2 f(x)\| \leq 2\|\nabla^2 f(x^*)\|$.

Para provar a desigualdade *ii)* tem de provar-se que $\|I - \nabla^2 f(x^*)^{-1}\nabla^2 f(x)\| \leq \frac{1}{2}$, para todo $x \in B_\delta(x^*)$. Assim,

$$\|I - \nabla^2 f(x^*)^{-1}\nabla^2 f(x)\| = \|\nabla^2 f(x^*)^{-1}(\nabla^2 f(x^*) - \nabla^2 f(x))\| \leq \overline{\beta}\,\overline{K}\|x^* - x\|$$

pelas Hipóteses 2.3 e 2.4. Se $\delta = \min\{r, \frac{1}{2\overline{\beta}\,\overline{K}}\}$, obtém-se $\|I - \nabla^2 f(x^*)^{-1}\nabla^2 f(x)\| \leq \frac{1}{2}$, e usando o Teorema de Banach B.5 no Apêndice B, vem

$$\|\nabla^2 f(x)^{-1}\| \leq \frac{\|\nabla^2 f(x^*)^{-1}\|}{1 - \|I - \nabla^2 f(x^*)^{-1}\nabla^2 f(x)\|} \leq 2\|\nabla^2 f(x^*)^{-1}\|.$$

Para provar a última desigualdade considera-se o Teorema 2.2 e a desigualdade *i)*. Para $x \in B_\delta(x^*)$, $x^* + t(x - x^*) \in B_\delta(x^*)$ para todo $t \in [0,1]$

$$\begin{aligned} \|\nabla f(x) - \nabla f(x^*)\| &\leq \int_0^1 \|\nabla^2 f(x^* + t(x - x^*))(x - x^*)\|\,dt \\ \|\nabla f(x)\| &\leq \int_0^1 2\|\nabla^2 f(x^*)\|\|x - x^*\|\,dt \\ &\leq 2\|\nabla^2 f(x^*)\|\|x - x^*\|. \end{aligned}$$

■

Estão reunidas agora as condições para começar o estudo local da convergência. Primeiro vai provar-se que a sucessão $\{x_k\}$ gerada pelo método de Newton modificado é bem definida.

Lema 2.2 *Considerem-se as Hipóteses 2.1, 2.2, 2.3 e 2.4. Então existe uma constante $\delta > 0$ com $x_0 \in B_\delta(x^*)$ tal que a sucessão $\{x_k\}$ gerada por (2.1), (2.2) e (2.4) é bem definida.*

Prova. O lema vai ser provado para as três situações separadamente.
Quando $w(x_k) = 0$, devido a (2.2), o iterando x_{k+1} é gerado a partir de

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

que é a iteração Newton, e $\|\nabla^2 f(x_k)^{-1}\| \leq 2\overline{\beta}$, usando a condição *ii)* do Lema 2.1, se $x_k \in B_\delta(x^*)$.

Quando $w(x_k)$ é 1, o algoritmo define o iterando da forma seguinte

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

que também é bem definido.

Finalmente quando $w(x_k) \in (0,1)$, $\nabla^2 f(x_k)$ é não singular para $x_k \in B_\delta(x^*)$ e o iterando é obtido a partir de

$$x_{k+1} = x_k - \left[w(x_k)\, I + (1 - w(x_k))\nabla^2 f(x_k)^{-1}\right] \nabla f(x_k)$$

em que

$$\begin{aligned} \left\| w(x_k)I + (1 - w(x_k))\nabla^2 f(x_k)^{-1} \right\| &\leq w(x_k) + (1 - w(x_k)) \parallel \nabla^2 f(x_k)^{-1} \parallel \\ &\leq w(x_k) + 2(1 - w(x_k)) \parallel \nabla^2 f(x^*)^{-1} \parallel \\ &\leq w(x_k) + 2(1 - w(x_k))\, \overline{\beta} \\ &< 1 + 2\overline{\beta} \end{aligned}$$

o que prova que x_{k+1} é bem definido se $x_k \in B_\delta(x^*)$. ■

A seguir vai ser estudada a razão de convergência local do método. O sucesso do método de Newton modificado depende de $w(x)$. Isto é confirmado pelo Lema 2.3 e Teorema 2.4. Começa-se por provar que o método é pelo menos q-linearmente convergente.

Teorema 2.3 *Considere-se $x_0 \in B_r(x^*)$. Usando as condições do Lema 2.2, a sucessão $\{x_k\}$ gerada por (2.1), (2.2) e (2.4) converge para $w(x_k) \in [0,1)$ pelo menos q-linearmente para x^* e obedece a*

$$\parallel e_{k+1} \parallel \leq \overline{\beta}\,\overline{K} \parallel e_k \parallel^2 + 2w(x_k) \parallel \nabla^2 f(x^*) \parallel (1 + 2\overline{\beta}) \parallel e_k \parallel \tag{2.5}$$

e

$$\parallel e_{k+1} \parallel < \parallel e_k \parallel \tag{2.6}$$

em que

$$e_k = x_k - x^*.$$

Prova. Seja $\delta > 0$ e $x_k \in B_\delta(x^*)$. Vai mostrar-se que na iteração k se verifica (2.5) para $w(x_k) \in [0,1)$, que

$$\parallel x_{k+1} - x^* \parallel < \parallel x_k - x^* \parallel$$

e consequentemente $x_{k+1} \in B_\delta(x^*)$.

Começa-se por assumir que $w(x_k) \in (0,1)$. De (2.1) e (2.2)

$$x_{k+1} = x_k - \left[w(x_k)\, I + (1 - w(x_k))\nabla^2 f(x_k)^{-1}\right] \nabla f(x_k).$$

Para estudar a convergência é útil colocar esta equação em função do erro. Seja $e_k = x_k - x^*$ o erro na iteração k, então

$$x_{k+1} - x^* = x_k - x^* - \nabla^2 f(x_k)^{-1}\nabla f(x_k) - w(x_k)\,\nabla f(x_k) + w(x_k)\nabla^2 f(x_k)^{-1}\nabla f(x_k)$$

ou, tomando normas e usando as Hipóteses 2.1, 2.2, 2.3 e 2.4 tem-se

$$\begin{aligned} \|e_{k+1}\| &\leq \overline{\beta}\,\overline{K} \parallel e_k \parallel^2 + w(x_k) \parallel \nabla f(x_k) \parallel + w(x_k) \parallel \nabla^2 f(x_k)^{-1} \parallel\parallel \nabla f(x_k) \parallel \\ &\leq \overline{\beta}\,\overline{K} \parallel e_k \parallel^2 + 2w(x_k) \parallel \nabla^2 f(x^*) \parallel\parallel e_k \parallel + 4w(x_k)\overline{\beta} \parallel \nabla^2 f(x^*) \parallel\parallel e_k \parallel \\ &\leq \overline{\beta}\,\overline{K} \parallel e_k \parallel^2 + 2w(x_k) \parallel \nabla^2 f(x^*) \parallel (1 + 2\overline{\beta}) \parallel e_k \parallel, \end{aligned} \tag{2.7}$$

o que prova (2.5).

Seja k_1 uma quantidade maior ou igual a um tal que

$$\parallel \nabla^2 f(x^*) \parallel \leq \frac{k_1}{\overline{\beta}} \tag{2.8}$$

para todo $x_k \in B_\delta(x^*)$ e considere-se

$$\delta = min\left\{r, \frac{1}{2\overline{\beta}\,\overline{K}}(1 - \frac{4k_1 w(x_k)}{\overline{\beta}}(1 + 2\overline{\beta}))\right\}. \tag{2.9}$$

Combinando (2.7) e (2.8) obtém-se

$$\begin{aligned} \parallel e_{k+1} \parallel &\leq \overline{\beta}\,\overline{K} \parallel e_k \parallel^2 + 2\frac{k_1}{\overline{\beta}} w(x_k)(1 + 2\overline{\beta}) \parallel e_k \parallel \\ &= \parallel e_k \parallel \left[\overline{\beta}\,\overline{K} \parallel e_k \parallel + 2\frac{k_1}{\overline{\beta}} w(x_k)(1 + 2\overline{\beta})\right]. \end{aligned}$$

De (2.9) e da expressão anterior

$$\begin{aligned} \parallel e_{k+1} \parallel &\leq \parallel e_k \parallel \left[\frac{1}{2}(1 - \frac{4k_1 w(x_k)}{\overline{\beta}}(1 + 2\overline{\beta})) + \frac{2k_1}{\overline{\beta}} w(x_k)(1 + 2\overline{\beta})\right] \\ &\leq \frac{1}{2} \parallel e_k \parallel \end{aligned}$$

provando (2.6) para o caso em que $0 < w(x_k) < 1$.

Para $w(x_k) = 0$, (2.7) reduz-se a $\| e_{k+1} \| \leq \overline{\beta}\,\overline{K} \| e_k \|^2$ e para $x_k \in B_\delta(x^*)$, assumindo que $\delta = min\left\{r, \frac{1}{2\overline{\beta}\,\overline{K}}\right\}$ chega-se a

$$\| e_{k+1} \| \leq \| e_k \| \left(\overline{\beta}\,\overline{K}\delta\right) \leq \frac{1}{2} \| e_k \|$$

conforme a prova do Teorema 5.2.1 em [28]. ■

Finalmente, o algoritmo atribui a $w(x_k)$ o valor 1 para resolver o problema da singularidade da matriz $\nabla^2 f(x_k)$. As equações (2.1) e (2.2) reduzem-se à descida máxima

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

O método da descida máxima tem uma razão de convergência linear e depende do número de condição da matriz $\nabla^2 f(x_k)$ (se esta matriz for mal condicionada a convergência torna-se mais lenta). Para mais detalhes veja-se [108].

Definição 2.3 *Seja $\{x_k\}$ uma qualquer sucessão que converge para x^*. Dadas duas funções reais de variável real, não negativas e contínuas, $g(x)$ e $h(x)$, diz-se que*

$$g(x_k) = \mathcal{O}(h(x_k))$$

quando $k \to \infty$, se $\lim_{k\to\infty} \frac{g(x_k)}{h(x_k)} < \infty$.

Para provar a razão q-quadrática para $0 \leq w(x_k) < 1$ vai ser preciso o seguinte lema.

Lema 2.3 *Considerem-se as condições do Lema 2.1 satisfeitas e seja*

$$h^T \nabla^2 f(x_k)^{-1} h \leq \mu h^T h,\ \forall x_k \in B_\delta(x^*),\ h \in \mathbb{R}^n \tag{2.10}$$

em que $\mu < 1$ não é necessariamente positivo. Então a função $w(x_k) \in [0,1)$ e definida por (2.4) satisfaz

$$w(x_k) = \mathcal{O}(\|\nabla f(x_k)\|_2^2).$$

Prova. Por uma questão de simplicidade de notação, vai usar-se w_k, ∇f_k e $\nabla^2 f_k$ no lugar respectivamente de $w(x_k)$, $\nabla f(x_k)$ e $\nabla^2 f(x_k)$. Da definição de w_k e para $0 < \varepsilon \leq 1$,

$$\begin{aligned} w_k &= \frac{\nabla f_k^T d_{N_k} + \varepsilon \|\nabla f_k\|_2^2}{\|\nabla f_k\|_2^2 + \nabla f_k^T d_{N_k}} \qquad (2.11) \\ &= \frac{-\nabla f_k^T \left(\nabla^2 f_k\right)^{-1} \nabla f_k + \varepsilon \|\nabla f_k\|_2^2}{\|\nabla f_k\|_2^2 - \nabla f_k^T \left(\nabla^2 f_k\right)^{-1} \nabla f_k} \\ &\leq \frac{\left\|\varepsilon I - \left(\nabla^2 f_k\right)^{-1}\right\| \|\nabla f_k\|_2^2}{\nabla f_k^T (I - \left(\nabla^2 f_k\right)^{-1}) \nabla f_k} \\ &\leq \frac{(\varepsilon + 2\overline{\beta}) \|\nabla f_k\|_2^2}{\nabla f_k^T (I - \left(\nabla^2 f_k\right)^{-1}) \nabla f_k} \qquad (2.12) \end{aligned}$$

usando a desigualdade $ii)$ do Lema 2.1. Seja vp_i os valores próprios de $\left(\nabla^2 f_k\right)^{-1}$ e v_i os correspondentes vectores próprios ortogonais. Como $\left(\nabla^2 f\right)^{-1}$ é simétrica $\{v_i\}$, $i = 1, \ldots, n$, é uma base de $\mathbb{R}^n$ podendo o vector $\nabla f_k \in \mathbb{R}^n$ ser definido como

$$\nabla f_k = \sum_{i=1}^{n} \alpha_i v_i$$

em que pelo menos um α_i é não nulo (Teorema 3.5.3 em [28]). Devido à ortogonalidade dos $v_i's$ verifica-se

$$\begin{aligned} \nabla f_k^T \left(I - \left(\nabla^2 f_k\right)^{-1}\right) \nabla f_k &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j (1 - vp_i) \left(v_i\right)^T v_j \\ &= \sum_{i=1}^{n} \alpha_i^2 (1 - vp_i). \qquad (2.13) \end{aligned}$$

Como $vp_i \leq \mu \, (i = 1, ..., n)$ de (2.10), $-vp_i \geq -\mu \Leftrightarrow 1 - vp_i \geq 1 - \mu$ e

$$\sum_{i=1}^{n} \alpha_i^2 (1 - vp_i) \geq (1 - \mu) \sum_{i=1}^{n} \alpha_i^2.$$

Então

$$\nabla f_k^T \left(I - \left(\nabla^2 f_k\right)^{-1}\right) \nabla f_k \geq (1 - \mu) \sum_{i=1}^{n} \alpha_i^2$$

e (2.12) reduz-se a

$$\begin{aligned} w_k &\leq \frac{(\varepsilon + 2\overline{\beta})}{(1-\mu)\sum_{i=1}^{n} \alpha_i^2} \|\nabla f_k\|_2^2 \\ &= M \|\nabla f_k\|_2^2 \end{aligned} \tag{2.14}$$

em que $M = \dfrac{(\varepsilon + 2\overline{\beta})}{(1-\mu)\sum_{i=1}^{n} \alpha_i^2} > 0$, para $\mu < 1$, completando-se desta forma a prova. ■

Neste momento, pode provar-se o teorema principal estabelecendo-se condições para uma convergência q-quadrática do método de Newton modificado para $w(x_k) \in [0, 1)$.

Teorema 2.4 *Sob as condições do Teorema 2.3 e Lema 2.3 e para $w(x_k)$ em (2.4) existe $\delta > 0$ tal que para todo o $x_0 \in B_\delta(x^*)$, a sucessão $\{x_k\}$ gerada pelo método de Newton modificado (2.1) e (2.2), quando $0 \leq w(x_k) < 1$, converge q-quadraticamente para x^*.*

Prova. Os casos $w(x_k) \in (0, 1)$ e $w(x_k) = 0$ são tratados separadamente. A prova para $0 < w(x_k) < 1$ faz-se directamente de (2.5), (2.14) e da desigualdade *iii)* do Lema 2.1:

$$\begin{aligned} \|e_{k+1}\| &\leq \overline{\beta}\,\overline{K} \|e_k\|^2 + 2w(x_k) \left\|\nabla^2 f(x^*)\right\| (1 + 2\overline{\beta}) \|e_k\| \\ &\leq \overline{\beta}\,\overline{K} \|e_k\|^2 + 8M \left\|\nabla^2 f(x^*)\right\|^3 \|e_k\|^2 (1 + 2\overline{\beta}) \|e_k\| \\ &= \left(\overline{\beta}\,\overline{K} + 8M \left\|\nabla^2 f(x^*)\right\|^3 (1 + 2\overline{\beta}) \|e_k\|\right) \|e_k\|^2 . \end{aligned}$$

Quando $w(x_k) = 0$, de (2.5) vem

$$\|e_{k+1}\| \leq \overline{\beta}\,\overline{K} \|e_k\|^2 ,$$

verificando-se uma razão de convergência q-quadrática para $0 \leq w(x_k) < 1$. ■

Para a resolução de problemas de optimização não lineares sem restrições, apresentou-se uma modificação do método de Newton. Esta estratégia irá ser estendida aos problemas não lineares com restrições, tratados no Capítulo 5.

2.5 Tópicos sobre a convergência global

A convergência global do método de Newton modificado e o estudo de algumas propriedades da direcção Newton modificada, no âmbito da optimização não linear sem restrições,

foram o tema da tese de mestrado de Caiado [11] e do trabalho de Caiado e Fernandes [12]. Esta secção vai ser dedicada à apresentação de alguns resultados obtidos pelas autoras, nomeadamente o teorema que estabelece condições para a convergência global do método de Newton modificado e algumas das propriedades da direcção Newton modificada. Esta descrição vai ser feita de uma forma sucinta, pelo que se aconselha a consulta dos trabalhos [11] e [12] para um estudo mais pormenorizado.

Considere-se o seguinte teorema que prova que, sob certas condições, a convergência global do método de Newton modificado é pelo menos r-linear, e cuja demonstração se omite, por estar descrita em qualquer dos dois trabalhos referidos. O teorema é, de uma certa forma, semelhante ao teorema de convergência para o método Levenberg-Marquardt apresentado no livro de Ortega e Rheinboldt [88, pp 505].

Teorema 2.5 (Teorema 5 de [11, pp 15]) *Seja $f : D \subset \mathbb{R}^n \to \mathbb{R}$ uma função duas vezes continuamente diferenciável num conjunto aberto D. Suponha-se que existe um $x_0 \in D$ tal que $L^0 = \{x \in D \mid f(x) \leq f(x_0)\}$ é compacto e que f tem um número finito de pontos estacionários em L^0. Assuma-se ainda que $B : L^0 \to \mathbb{R}^{n\times n}$ em que $B(x) = (1 - w(x))\nabla^2 f(x)^{-1}$ com $w(x) \in [0, 1)$ (em (2.4)) é contínua para todo $x \in L^0$ e que*

$$\mu_0 h^T h \leq h^T B(x) h \leq \mu_1 h^T h, \quad \forall x \in L^0, h \in \mathbb{R}^n$$

$$h^T \nabla^2 f(x) h \leq \gamma_1 h^T h, \quad \forall x \in L^0, h \in \mathbb{R}^n$$

onde μ_0, μ_1 e γ_1 não são necessariamente todos positivos, mas ou $\gamma_1 \leq 0$ e $\mu_1 > 0$ ou $\gamma_1 > 0$ e $-\frac{\gamma_1}{2+\gamma_1} \leq \mu_1 \leq \frac{2}{\gamma_1} - 1$. Considere-se

$$x_{k+1} = x_k - [(1 - w(x_k))\nabla^2 f(x_k)^{-1} + w(x_k) I]\nabla f(x_k), \quad k = 0, 1, 2, \ldots$$

com

$$\frac{1}{2}\widehat{\gamma}_1 - \mu_0 < \eta_0 \leq w(x_k) \leq \frac{1}{2}\widehat{\gamma}_1 + \frac{\mu_1}{\mu_1 + 1} < 1, \quad \widehat{\gamma}_1 = max(0, \gamma_1).$$

Então

$$\{x_k\} \subset L^0,$$

$$\lim_{k\to\infty} x_k = x^* \quad e \quad \nabla f(x^*) = 0$$

e se $\nabla^2 f(x^*)$ é não singular, a convergência é pelo menos r-linear.

Prova-se ainda que a matriz dos coeficientes associada ao método de Newton modificado, $(1 - w(x))\nabla^2 f(x)^{-1} + w(x)I$, é sob certas condições, definida positiva (ver Lema 6 de [11, pp 21]).

Relativamente à direcção Newton modificada apresentam-se algumas propriedades interessantes. Assim, a direcção Newton modificada, d_{NM}, sob certas condições:

(i) é uma direcção do tipo gradiente, no sentido de Ortega e Rheinboldt (Lema 7 em [11, pp 22]);

(ii) satisfaz uma fracção do decréscimo óptimo (Teorema 8 em [11, pp 24]) e como consequência induz uma redução significativa no modelo quadrático da função no iterando x_k;

(iii) não se torna ortogonal ao simétrico do gradiente (Lema 10 em [11, pp 32]).

2.6 Comentários, síntese e motivação

O método de Newton é um dos métodos iterativos mais conhecidos e usados na resolução de problemas de optimização com e sem restrições. Nos problemas sem restrições nas variáveis a sua implementação é equivalente à aproximação local da função objectivo por um modelo quadrático. Quando existem restrições nas variáveis a aplicação pode ser feita sobre uma função de penalidade ou de barreira, consoante existem restrições de igualdade e/ou desigualdade. Estas funções constroem-se adicionando à função objectivo um termo que penaliza a violação das restrições ou forma uma barreira na fronteira da região admissível. A solução do modelo quadrático gera uma aproximação à solução do problema original e o processo repete-se.

É possível garantir convergência desta sucessão de aproximações assim obtida se o processo iterativo for iniciado perto da solução, uma vez que certas propriedades só são verificadas numa pequena vizinhança da solução.

Uma propriedade importante tem a ver com o facto da matriz ser definida positiva ao longo do processo iterativo. Muito se tem investido nas modificações a introduzir à matriz dos coeficientes do sistema Newton por forma a garantir que ela se conserve definida positiva. Neste capítulo apresentou-se um novo algoritmo do tipo Newton, designado por método de Newton modificado. Foi feito um estudo sobre a convergência local e foram enunciadas as condições de convergência global, bem como algumas propriedades da direcção resultante do algoritmo. Algumas delas motivaram a aplicação deste método à resolução de problemas com restrições de igualdade numa perspectiva de técnicas de penalidade, como se verá no Capítulo 5. O método de Newton modificado já foi também

Newton	ov	13	6	s	s	15	25	s
Newton modificado	11+8	>50	6+2	3+3	9+27	13+1	6+2	8+4
Newton	20	30	60	>50	26	s	6	
Newton modificado	9+1	12+3	34+4	15+12	20+1	10+1	>50	

Tabela 2.1: Número de iterações

utilizado no trabalho [78] no âmbito da resolução de sistemas de equações não lineares. Para um universo de vinte e sete problemas, num total de quarenta e seis testes (com problemas com vários valores iniciais), retirados de [82] obteve-se 93% de sucessos contra 85% para o método de Newton. Apenas em quinze testes houve diferenças no desempenho dos respectivos algoritmos. Estes casos estão ilustrados na Tabela 2.1 onde é apresentado o número de iterações realizadas até convergir. Na linha do método de Newton modificado, o primeiro número corresponde a direcções Newton aceites e o segundo a direcções modificadas. Na tabela, "ov" significa que ocorreu *overflow* e "s" significa que se verificou singularidade da matriz do Jacobiano.

Capítulo 3

Técnicas de globalização

Na globalização dos métodos do gradiente podem implementar-se dois tipos de técnicas. A técnica de procura unidimensional e a baseada em regiões de confiança. A primeira pode, hoje em dia, ser implementada de acordo com estratégias não monótonas, iniciadas com a introdução do esquema de vigilância *watchdog* [14], e monótonas. Esta foi a técnica seleccionada para a globalização dos algoritmos neste trabalho. O segundo tipo de técnica, que surgiu com os trabalhos de Levenberg e Marquardt, tem já uma aplicação generalizada em problemas de optimização não linear. Neste capítulo faz-se uma descrição sumária destas estratégias de globalização.

3.1 Introdução

A maioria dos métodos de optimização baseados no gradiente possuem uma convergência do tipo local, ou seja, convergem para a solução se a estimativa inicial do processo iterativo estiver suficientemente próxima da solução, ou seja, pertencer à região de convergência. Nos métodos que convergem, independentemente da aproximação inicial x_0, a convergência diz-se global e os métodos globalmente convergentes. Uma vez que raramente se tem informação acerca da solução do problema, a selecção duma estimativa inicial na região de convergência, não é uma tarefa fácil. Surge então a necessidade de introduzir alterações aos métodos localmente convergentes, com o objectivo de alargar a região de convergência, tornando-os globalmente convergentes, mantendo no entanto as suas excelentes propriedades de convergência local. As alterações referidas são identificadas como técnicas de globalização. Em regiões afastadas da solução, aproximações baseadas nas séries de Taylor da função são pouco fiáveis e, neste caso, as estratégias de globalização têm uma parte activa no algoritmo, canalizando o processo iterativo na direcção da solução,

impedindo que o processo iterativo se afaste da solução ou mesmo divirja. Na região de convergência, perto da solução, estas estratégias embora disponíveis não são normalmente necessárias.

Existem dois esquemas fundamentais de indução de convergência global - procura unidimensional e regiões de confiança. Ambas as técnicas geram passos de procura, cada uma à sua maneira. A procura unidimensional usa um modelo para gerar a direcção de procura e em seguida tenta encontrar o comprimento do passo adequado ao longo dessa direcção. A técnica de regiões de confiança define uma região à volta da aproximação actual, dentro da qual o modelo quadrático parece ser uma representação adequada da função objectivo, escolhendo como passo o minimizante do modelo nessa região de confiança. Na prática a direcção e o comprimento do passo são calculados em simultâneo. Se um passo não pode ser aceite, o tamanho da região de confiança é reduzido e a técnica procura um novo minimizante. Em geral, a direcção muda sempre que o tamanho da região de confiança é alterado.

Estas duas técnicas de globalização de métodos são usadas com sucesso, em algoritmos de optimização, e nenhuma delas se tem mostrado superior à outra. Experiências computacionais têm mostrado que, em média, elas são comparáveis em termos de eficiência, apesar de, em determinados problemas o seu desempenho ser diferente. De seguida vai fazer-se uma descrição sumária destas técnicas.

3.2 Procura unidimensional

A ideia central desta técnica de globalização, depois de obtida, de alguma forma, uma direcção de procura, consiste no cálculo do comprimento do passo a tomar ao longo da direcção, de modo a gerar um próximo ponto com, no mínimo, menor valor da função objectivo. O sucesso desta técnica depende das escolhas feitas para a direcção e para o comprimento do passo.

O termo procura unidimensional surge devido ao facto da procura ser baseada numa função de apenas uma só variável. Idealmente, o objectivo seria calcular a solução de

$$\arg\, min\, \varphi(\alpha) \equiv f(x_k + \alpha d_k), \tag{3.1}$$

ou seja, α seria o minimizante global da função a uma variável $\varphi(\alpha)$. Este problema, mesmo para o cálculo dum minimizante local, quando resolvido exactamente torna-se nalguns casos inviável e pode ser muito pesado em termos de custos computacionais. Na prática aceita-se uma aproximação a (3.1) de tal forma que se verifique uma redução no valor da função objectivo com o mínimo de esforço computacional. Esta estratégia caracteriza uma procura monótona. Os algoritmos típicos de procura unidimensional

experimentam uma sucessão de possíveis valores para α aceitando o primeiro que satisfaça uma determinada condição de monotonia da função objectivo.

Uma forma de garantir a convergência resulta da imposição de certas condições à direcção de procura d_k e ao comprimento do passo α. Assim, a direcção terá de ser suficientemente descendente e ser do tipo gradiente. O valor de α deverá produzir uma redução significativa no valor da função objectivo e não deverá ser demasiado pequeno.

3.2.1 Algumas propriedades da direcção de procura

Quando d_k é quase ortogonal ao gradiente $\nabla f(x_k)$, embora sendo ainda uma direcção descendente, o progresso do algoritmo pode ser posto em causa. Para garantir que esta situação indesejável não ocorra assume-se que

$$-\frac{d_k^T \nabla f(x_k)}{\|d_k\| \cdot \|\nabla f(x_k)\|} \geq \eta_1 > 0, \tag{3.2}$$

para todo o k, em que $\eta_1 > 0$ é uma tolerância estabelecida (veja-se a Definição 2.2). Esta condição pode ser interpretada sob o ponto de vista do ângulo θ entre d_k e $-\nabla f(x_k)$, ou seja,

$$cos\,\theta \geq \eta_1 > 0$$

sendo conhecida também como condição do ângulo ([85, pp 314]).

Definição 3.1 *A direcção diz-se do tipo gradiente se*

$$\|d_k\| \geq m \|\nabla f(x_k)\| \tag{3.3}$$

para todo o k e $m > 0$.

Esta condição diz que a norma da direcção não deve ser muito menor do que a norma do gradiente. Estas duas condições (3.2) e (3.3) devem ser normalmente verificadas pelos algoritmos utilizados para o cálculo de direcções de procura.

3.2.2 Cálculo do comprimento do passo

Uma aproximação linear a $f(x_k + \alpha d_k)$ obtém-se a partir da expansão em série de Taylor de f no iterando x_k

$$l(x_k + \alpha d_k) = f(x_k) + \alpha\, d_k^T \nabla f(x_k). \tag{3.4}$$

Na procura unidimensional pretende-se que o comprimento do passo α origine uma redução no valor da função que seja pelo menos uma fracção da redução prevista no modelo linear (3.4), ou seja,

$$f(x_k + \alpha d_k) - f(x_k) \leq \mu_1 \alpha d_k^T \nabla f(x_k) \tag{3.5}$$

em que $0 < \mu_1 < 1$. Quando μ_1 está próximo de zero, esta condição é facilmente satisfeita, uma vez que a redução exigida no valor da função é pequena. A condição (3.5) é conhecida como condição de Armijo e exige uma redução considerada significativa na função $f(x)$, de uma iteração para a seguinte. Se α for pequeno, a aproximação linear será boa e a condição de redução significativa será satisfeita. Se α for grande, a redução dado pela aproximação linear pode ser muito diferente da redução efectiva e a condição seria violada pelo que, esta condição não permite a aceitação de grandes valores para α.

A condição (3.5) não é por si só suficiente para garantir o progresso do algoritmo. Para se evitarem pequenos valores para α, introduz-se uma segunda imposição chamada condição de curvatura que obriga a que α satisfaça

$$\nabla f(x_k + \alpha d_k)^T d_k \geq \eta_2 \nabla f(x_k)^T d_k \tag{3.6}$$

para alguma constante $\eta_2 \in (\mu_1, 1)$, sendo μ_1 a constante de (3.5). As condições de Armijo (3.5) e de curvatura (3.6) em conjunto são conhecidas pelas condições de Wolfe-Armijo. São condições independentes da escala e usadas em muitos métodos de procura unidimensional. A prova de que, para qualquer função continuamente diferenciável e limitada inferiormente, existe um valor de α que satisfaça as condições de Wolfe-Armijo, pode ser consultada em [87, pp 40].

Da mesma forma que as condições de Wolfe-Armijo, o princípio de Goldstein

$$\mu_2 \alpha \nabla f(x_k)^T d_k \leq f(x_k + \alpha d_k) - f(x_k) \leq \mu_1 \alpha \nabla f(x_k)^T d_k \tag{3.7}$$

em que $0 < \mu_1 < \mu_2 < 1$, também exige uma redução significativa na função objectivo e impede que o valor de α seja muito pequeno ([87]). As duas desigualdades (3.7) são também conhecidas pelas condições de Goldstein-Armijo. Estas, quando comparadas com as condições de Wolfe-Armijo, apresentam a desvantagem da primeira desigualdade poder excluir todos os minimizantes de $\varphi(\alpha) \equiv f(x_k + \alpha d_k)$. No entanto, ambas têm muito em comum e as suas teorias de convergência são muito semelhantes (veja-se por exemplo ([87]).

A condição de Armijo por si só não é um garante do progresso do algoritmo ao longo de determinada direcção. Se se introduzir um esquema apropriado de redução dos valores de α (*backtracking*) (veja-se o Algoritmo 6.3.5 em [28, pp 126]), a condição de curvatura

(3.6) pode ser dispensada, usando-se apenas a condição de Armijo ([87]). Numa tentativa inicial, escolhe-se um valor $\overline{\alpha}$ (normalmente um) e caso não verifique (3.5) vão sendo experimentados valores de uma sucessão que converge para zero, $\left\{\overline{\alpha}\beta^k\right\}$ $k = 1, 2, \ldots$ com $\beta < 1$, até que a condição se verifique. Para evitar que α atinja valores demasiados pequenos, é estabelecido um limite para o passo, por exemplo $0 < \delta_1 \leq \|\alpha d_k\|_2$, em que δ_1 é a distância mínima permitida entre duas aproximações sucessivas x_k e x_{k+1}.

Uma escolha adequada de $\overline{\alpha}$ pode ser feita da forma que a seguir se mostra. Se na minimização de $f(x)$ se considerar $\varphi(\alpha) = f(x_k + \alpha d_k)$ então α é uma raiz de $\varphi'(\alpha) = f'_\alpha(x_k + \alpha d_k) = 0$, ou seja, α é a solução da seguinte equação não linear unidimensional

$$\nabla f(x_k + \alpha d_k)^T d_k = 0. \tag{3.8}$$

A equação (3.8) pode ter várias raízes sendo importante saber qual delas deve ser escolhida. Segundo Curry, ela deve ser a menor raiz positiva, partindo do pressuposto de que $\nabla f(x_k)^T d_k < 0$ ([88]).

Directamente relacionado com o princípio de Curry surge o princípio de Altman segundo o qual, para um valor fixo de $\eta_2 \in [0, 1)$, α é a menor raiz positiva da equação ([52])

$$\left[\nabla f(x_k + \alpha d_k) - \eta_2 \nabla f(x_k)\right]^T d_k = 0. \tag{3.9}$$

Se $\eta_2 = 0$ a equação anterior reduz-se à equação (3.8); se $\eta_2 > 0$ o princípio de Altman interpreta α como o menor valor positivo para o qual $\varphi'(\alpha)$ é igual à fracção η_2 do valor original $\varphi'(0)$, isto é, $x_{k+1} = x_k + \alpha d_k$ é o ponto mais próximo de x_k que verifica (3.9).

A resolução de (3.9) pode ser feita aproximando-se o vector $\nabla f(x_{k+1})$ por um modelo linear.

Considere-se a seguinte definição.

Definição 3.2 *Uma função $F : X \to Y$ tem derivada Fréchet $F'(x) \in \mathcal{L}(X, Y)$, conjunto de matrizes do espaço $X \times Y$, com $x \in X$ se*

$$F(y) = F(x) + F'(x)(y - x) + o(\| y - x \|)) \tag{3.10}$$

em que $\lim_{y \to x} \frac{o(\|y-x\|)}{\|y-x\|} = 0$ *([65]).*

Assim, se $X = \mathbb{R}^n$, $Y = \mathbb{R}^n$, e $F = \nabla f(x) : \mathbb{R}^n \to \mathbb{R}^n$, aplicando (3.10) a $\nabla f(x_{k+1})$, vem

$$\nabla f(x_{k+1}) = \nabla f(x_k) + \alpha \nabla^2 f(x_k) d_k + o(\| \alpha d_k \|)$$

com $\lim_{x_{k+1}\to x_k} \frac{o(\| \alpha d_k \|)}{\| \alpha d_k \|} = 0.$

A aproximação linear a $\nabla f(x_{k+1})$ é

$$\nabla f(x_{k+1}) \approx \nabla f(x_k) + \alpha \nabla^2 f(x_k) d_k \tag{3.11}$$

e considerando $z_k = x_k + d_k$ ($\alpha = 1$) tem-se também

$$\nabla f(z_k) \approx \nabla f(x_k) + \nabla^2 f(x_k) d_k.$$

Usando o resultado anterior em (3.11) ([38]) fica-se com

$$\nabla f(x_{k+1}) \approx \nabla f(x_k) + \alpha \left[\nabla f(z_k) - \nabla f(x_k)\right]$$

que substituindo em (3.9) origina

$$\left[\nabla f(x_k) + \alpha \left[\nabla f(z_k) - \nabla f(x_k)\right] - \eta_2 \nabla f(x_k)\right]^T d_k = 0.$$

Resolvendo em ordem a α obtém-se uma estimativa de um valor adequado para α,

$$\alpha = \frac{(\eta_2 - 1)\nabla f(x_k)^T d_k}{[\nabla f(z_k) - \nabla f(x_k)]^T d_k}. \tag{3.12}$$

Em termos de implementação esta estratégia calcula um ponto auxiliar $z_k = x_k + d_k$ que, com o obtido a partir de α calculado por (3.12), escolhe o que tem menor valor de $f(x)$. O ponto seleccionado vai verificar a condição de Armijo (3.5) introduzindo, se necessário, um esquema de redução de α. A técnica que incorpora a equação (3.12) baseada nos princípios de Curry e Altman juntamente com a condição de Armijo é designada por critério de Curry-Altman-Armijo (CAA). O Algoritmo 3.1 descreve este procedimento.

De seguida vão ser enunciadas as condições que garantem que o comprimento do passo gerado pelo Algoritmo 3.1, que também inclui uma estratégia de redução de α, satisfaz, em cada iteração, uma das condições:

$$i) \qquad f(x_k + \alpha_k d_k) - f(x_k) \leq -\xi_1 \frac{(\nabla f(x_k)^T d_k)^2}{\|d_k\|^2} \tag{3.13}$$

$$ii) \qquad f(x_k + \alpha_k d_k) - f(x_k) \leq \xi_2 \nabla f(x_k)^T d_k \tag{3.14}$$

em que ξ_1 e ξ_2 são constantes positivas. A primeira condição é razoavelmente conhecida e corresponde a um caso especial do princípio da redução significativa (*principle of sufficient decrease*) de acordo com a obra de Ortega e Rheinboldt [88, pp 479].

Algoritmo 3.1 (Procura unidimensional Curry-Altman-Armijo)

Dados μ_1, η_2 *e* δ_1;
{*calcular* $z_k = x_k + d_k$, $f(z_k)$ *e* $\nabla f(z_k)$}
{*calcular* α *através de (3.12)*}
{*calcular* $x = x_k + \alpha d_k$ *e* $f(x)$}
Se $(f(z_k) < f(x))$ ***Então***
$\{x = z_k; \alpha = 1\}$
FimSe
Enquanto $\left(f(x_k) - f(x) \leq -\mu_1 \alpha \nabla f(x_k)^T d_k\right)$ ***Fazer***
Se $(\|\alpha d_k\|_2 > \delta_1)$ ***Então***
$\left\{\alpha = \dfrac{\alpha}{2}; \quad x = x_k + \alpha d_k\right\}$
Senão
$\{\alpha = 1; \quad x = x_k + \alpha d_k\}$
$\{SAIR\}$
FimSe
FimEnquanto
$\{x_{k+1} = x\}$

Considere-se primeiro a hipótese.

Hipótese 3.1 *i) A função* $f(x)$ *é duas vezes continuamente diferenciável;*
ii) O conjunto de nível $L^0 = \{x : f(x) \leq f(x_0)\}$ *é convexo e existem constantes positivas* m_1 *e* m_2 *tais que*

$$m_1 \|h\|_2^2 \leq h^T \nabla^2 f(x) h \leq m_2 \|h\|_2^2$$

para todo o $h \in \mathbb{R}^n$ *e todo o* $x \in L^0$.

Seguindo a ideia do Lema 4.1 de [10, pp 737] tem-se:

Teorema 3.1 *De acordo com a Hipótese 3.1 existem constantes positivas* ξ_1 *e* ξ_2 *tais que, para todo o* x_k *e* d_k *que verificam* $\nabla f(x_k)^T d_k < 0$, *o comprimento do passo* α_k *gerado pelo Algoritmo 3.1 satisfaz uma das condições (3.13) ou (3.14).*

Prova. Se a condição de Armijo

$$f(x_k + \alpha d_k) \leq f(x_k) + \mu_1 \alpha \nabla f(x_k)^T d_k, \quad \text{com } 0 < \mu_1 < 1$$

é verificada para $\alpha_k = 1$, então

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \xi_2 \nabla f(x_k)^T d_k, \quad \text{com } \xi_2 = \mu_1.$$

Por sua vez, se a condição de Armijo é verificada para α_k calculado pela fórmula

$$\alpha_k = \frac{(\eta_2 - 1)\nabla f(x_k)^T d_k}{(\nabla f(x_k + d_k) - \nabla f(x_k))^T d_k}, \quad \text{com } 0 < \eta_2 < 1 \tag{3.15}$$

tem-se, pelo Teorema B.3 no Apêndice B, com $0 < \theta < 1$, e de acordo com a Hipótese 3.1

$$\begin{aligned} f(x_k + \alpha_k d_k) - f(x_k) &\leq \mu_1 \frac{(\eta_2 - 1)\left(\nabla f(x_k)^T d_k\right)^2}{d_k^T \nabla^2 f(x_k + \theta d_k) d_k} \\ &\leq \frac{-\mu_1(1 - \eta_2)\left(\nabla f(x_k)^T d_k\right)^2}{m_1 \|d_k\|_2^2} \end{aligned}$$

que origina (3.13) com $\xi_1 = \dfrac{\mu_1(1 - \eta_2)}{m_1}$.

Suponha agora que α_k ou é menor do que um ou menor do que o valor calculado por (3.15). Nesta situação a condição de Armijo não foi verificada para um valor de α, $\overline{\alpha}$, tal que $\overline{\alpha} \leq 2\alpha_k$ e

$$f(x_k + \overline{\alpha} d_k) - f(x_k) > \mu_1 \overline{\alpha} \nabla f(x_k)^T d_k.$$

Usando o Teorema B.3 do Apêndice B com $0 < \theta < 1$, obtém-se

$$\overline{\alpha} \nabla f(x_k + \theta \overline{\alpha} d_k)^T d_k > \mu_1 \overline{\alpha} \nabla f(x_k)^T d_k$$

ou

$$\overline{\alpha} \left[\nabla f(x_k + \theta \overline{\alpha} d_k) - \nabla f(x_k)\right]^T d_k > (\mu_1 - 1)\, \overline{\alpha} \nabla f(x_k)^T d_k.$$

De acordo com a Hipótese 3.1 e usando novamente o Teorema B.3 do Apêndice B com $0 < \theta_1 < 1$, tem-se

$$\overline{\alpha}^2 m_2 \|d_k\|_2^2 \geq \overline{\alpha}^2 \theta d_k^T \nabla^2 f(x_k + \theta_1 \theta \overline{\alpha} d_k) d_k > (\mu_1 - 1)\, \overline{\alpha} \nabla f(x_k)^T d_k$$

que implica

$$\alpha_k \geq \frac{1}{2}\overline{\alpha} > \frac{-(1 - \mu_1)\nabla f(x_k)^T d_k}{2 m_2 \|d_k\|_2^2}.$$

Substituindo esta expressão na condição de Armijo, chega-se a

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \frac{-\mu_1(1-\mu_1)\left(\nabla f(x_k)^T d_k\right)^2}{2m_2 \|d_k\|_2^2}$$

que, mais uma vez, verifica (3.13) com $\xi_1 = \frac{\mu_1(1-\mu_1)}{2m_2}$. ■

Se a redução do valor da função objectivo é exigida em todas as iterações, a procura diz-se monótona. Se o algoritmo de procura não obrigar a uma redução em todas as iterações, ou seja, permitir aumentos controlados da função objectivo em algumas iterações, a procura diz-se não monótona.

3.2.3 Procura não monótona

Como já foi referido, a convergência global pode ser forçada através de esquemas de procura unidimensional que obriguem a uma redução significativa de determinada função. No entanto, quando se está muito próximo da solução x^*, os esquemas de imposição da redução da função objectivo podem obrigar a que não seja aceite um comprimento de passo unitário ($\alpha = 1$), pondo em causa a razão de convergência local do método. Esta situação foi apresentada pela primeira vez por Maratos na sua tese de doutoramento, ficando a ser conhecida pelo "efeito de Maratos". Uma das formas de contornar esta situação indesejável foi proposta por Chamberlain, Powell, Lemarechal e Pedersen em [14], introduzindo um esquema de vigilância (*watchdog*), no qual não existe uma obrigatoriedade permanente de redução da função, preservando-se a convergência global. Este esquema de vigilância permite que em certas iterações se use um critério relaxado, menos restritivo, em vez do critério de referência, para decidir se um passo é ou não aceite. Para estes autores e nalguns casos, a procura não monótona parece apresentar vantagens relativamente à monótona.

Vários outros autores apresentam estratégias de procura não monótona nos seus trabalhos, [61], [62], [110], [40] e [57]. Grippo, Lampariello e Lucidi em [61] baseados no trabalho [14] utilizam uma abordagem tipo esquema de vigilância, relaxando algumas condições, de forma a conservar a convergência global. Neste trabalho os autores impõem que o valor da função em cada nova iteração, satisfaça a condição de Armijo tendo como referência o valor máximo da função para um número prédeterminado de iterações anteriores. Consiste numa generalização do critério de Armijo, no sentido de que permite acréscimos no valor da função sem que as propriedades de convergência sejam postas em causa. Em termos de resultados experimentais, os autores salientam que o uso desta técnica de procura unidimensional não monótona, em conjunto com o método de Newton, apresenta melhorias

principalmente nos estágios intermédio e final do processo de minimização. Sugerem ainda, a possibilidade de aplicar uma estratégia de procura não monótona aos métodos baseados em regiões de confiança. Na sequência deste trabalho, os mesmos autores em [62] apresentam uma estratégia de estabilização onde combinam a técnica de procura unidimensional não monótona de [61] com um teste ao tamanho de $\|d_k\|$ definindo pontos de referência e pontos de controlo e utilizando determinados critérios de aceitação do passo de procura. Discutem ainda algumas propriedades de convergência do método tipo Newton, concluindo com a análise de convergência do algoritmo e apresentando resultados numéricos promissores.

Zhu em [110] utiliza uma abordagem de procura unidimensional não monótona no âmbito dos métodos Quasi-Newton projectados, evidenciando vantagens significativas nas iterações intermédias e finais do processo de minimização. Sugere também a aplicação de estratégias não monótonas no âmbito das regiões de confiança.

Ferris, Lucidi e Roma em [40] apresentam um algoritmo para optimização sem restrições que incorpora uma procura curvilínea baseada numa direcção Newton modificada e noutra de curvatura negativa e a estratégia de estabilização não monótona já referida.

Em [57] Gould, Lucidi, Roma e Toint fazem referência ao mesmo algoritmo de procura unidimensional não monótona, no âmbito da optimização sem restrições. Segundo os autores, o modelo usado para representar a função objectivo "retém em memória" o valor da função nas últimas iterações, com a finalidade de tornar o algoritmo mais abrangente, ou seja, menos "míope" nas palavras dos autores. Desta forma, o comportamento do algoritmo não é completamente dominado pela natureza local da função objectivo, tendo-se uma maior visão global do comportamento da mesma.

O Algoritmo 3.2 contém a estratégia não monótona na sua versão básica. O parâmetro M denota o número de iterações em que é permitido um aumento do valor da função objectivo. $M = 5$ tem sido o valor mais usado nos trabalhos atrás referidos. Para $M = 0$, o algoritmo reduz-se ao critério de Armijo (procura monótona) com o esquema de redução de α. O parâmetro N indica quantas iterações são feitas no início do processo com procura monótona. Os valores que têm originado melhores resultados são $N = 2$ e $N = 3$. Em cada conjunto de M iterações já realizadas selecciona-se o maior valor de $f(x)$ e coloca-se em f_{max}. A verificação do critério de Armijo é feita, através da comparação do valor da função na iteração corrente com o valor, não da iteração anterior mas com f_{max}. Desta forma não se exige uma redução de $f(x)$ em todas as iterações.

Algoritmo 3.2 (Versão básica da estratégia não monótona)

Dados M, N, $\delta_1; i = 1;$
(Nas* N *primeiras iterações a procura é monótona;*
f_{max} *é o valor máximo de* $f(x)$ *nas últimas* M *iterações *)*
Se $(i \leq N)$ ***Então***
  $\{f_{max} = f(x_k)\}$
Senão
  Se $(i > (M + N))$ ***Então***
    $\{ind = mod(i - N - 1, M) + 1\}$
    $\{lista(ind) = f(x_k)\}$
    $\{f_{max} = MAXVAL(lista)\}$
  Senão
    Se $(i = N + 1)$ ***Então***
      $f_{max} = f(x_k)$
    FimSe
    $lista(i - N) = f(x_k)$
    Se $(f(x_k) > f_{max})$ ***Então***
      $f_{max} = f(x_k)$
    FimSe
  FimSe
FimSe
$\{\alpha = 1;\ x = x_k + \alpha d_k\}$
Enquanto $\left(f_{max} - f(x) \leq -\mu_1\, \alpha \nabla f(x_k)^T d_k\right)$ ***Fazer***
  Se $(\|\alpha d_k\|_2 > \delta_1)$ ***Então***
    $\left\{\alpha = \dfrac{\alpha}{2};\quad x = x_k + \alpha d_k\right\}$
  Senão
    $\{\alpha = 1;\quad x = x_k + \alpha d_k\}$
    $\{SAIR\}$
  FimSe
FimEnquanto
$\{x_{k+1} = x; i = i + 1;\}$

Facchinei, Júdice e Soares utilizam também uma procura não monótona de estabilização num método de Newton baseado numa estratégia de restrições activas, em problemas específicos de grandes dimensões com restrições de limites simples (veja-se em [35]).

O Algoritmo 3.3 apresenta uma outra versão não monótona implementada neste trabalho. A principal diferença relativamente ao algoritmo anterior, reside na definição do valor de f_{max}, que é actualizado uma vez em cada M iterações, não com o valor máximo das últimas M iterações mas com o valor de $f(x)$ na iteração final do conjunto de M iterações.

Algoritmo 3.3 (Procura unidimensional monótona de M em M)

Dados M, N, δ_1; $i = 1$;
(Nas N primeiras iterações a procura é monótona e em cada M iterações* f_{max} *é actualizado *)*
Se $((i \leq N)$ *OU* $(mod(i+N, M) = 1))$ ***Então***
  $\{f_{max} = f(x_k)\}$
FimSe
$\{\alpha = 1;\ x = x_k + \alpha d_k\}$
Enquanto $(f_{max} - f(x) \leq -\mu_1 \alpha \nabla f(x_k)^T d_k)$ ***Fazer***
  Se $(\|\alpha d_k\|_2 > \delta_1)$ ***Então***
    $\left\{\alpha = \frac{\alpha}{2};\quad x = x_k + \alpha d_k\right\}$
  Senão
    $\{\alpha = 1;\quad x = x_k + \alpha d_k\}$
    $\{SAIR\}$
  FimSe
FimEnquanto
$\{x_{k+1} = x; i = i + 1\}$

3.3 Regiões de confiança

A outra estratégia de globalização baseia-se nas regiões de confiança. A ideia original surgiu com os trabalhos de Levenberg em 1944 e Marquardt em 1963 para a resolução de problemas de mínimos quadrados não lineares (veja-se em [81] e [108]) e tem sido adaptada desde então, aos mais diversos tipos de problemas de optimização (sem restrições [98], com restrições de limites simples [17], com restrições de igualdade [13], [23], [29] e [33], só para referenciar alguns), aos problemas de controlo óptimo [25], à resolução de sistemas não lineares formulados como problemas de mínimos quadrados [24] e à programação semi-infinita [100]. São ainda importantes os trabalhos [7], [64] e [106]. Uma vez que esta

estratégia não foi utilizada neste trabalho, far-se-á apenas uma breve introdução ao tema, reportando para a bibliografia um estudo mais detalhado.

Considere-se o modelo quadrático da função $f(x)$, na iteração k,

$$m_k(d) = f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T B(x_k) d$$

em que d é a direcção de procura e $B(x_k)$ uma matriz simétrica. Do Teorema B.4 no Apêndice B tem-se

$$f(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_k + td) d$$

para algum $t \in (0,1)$, e como $m_k(d) = f(x_k) + \nabla f(x_k)^T d + \mathcal{O}(\|d\|^2)$, a diferença entre $m_k(d)$ e $f(x_k + d)$ é $\mathcal{O}(\|d\|^2)$. O erro da aproximação é pequeno se d for pequeno.

Quando $B(x_k)$ é a matriz Hessiana $\nabla^2 f(x_k)$, o modelo coincide com os três primeiros termos da série de Taylor, sendo o erro $\mathcal{O}(\|d\|^3)$ e neste caso, aproxima com bastante precisão a função quando d é pequeno. Na situação em que $B(x_k) = \nabla^2 f(x_k)$ surge o método de Newton. Nesta altura, vai considerar-se $B(x_k)$ genérica, assumindo apenas as propriedades de simetria e de uniformemente limitada.

A solução do problema

$$\begin{array}{ll} min & m_k(d) = f(x_k) + \nabla f(x_k)^T d + \tfrac{1}{2} d^T B(x_k) d \\ d \in \mathbb{R}^n & sujeito \quad a \quad \|d\| \leq \Delta_k, \end{array} \tag{3.16}$$

em que Δ_k é o raio da região de confiança, é o deslocamento efectuado na iteração k. A abordagem de regiões de confiança obriga à resolução de uma sequência de subproblemas do tipo (3.16) nos quais tanto a função objectivo como a restrição ($d^T d \leq \Delta_k^2$) são quadráticas. Esta tarefa não é fácil. Quando a direcção d não satisfaz a restrição, é necessário determinar λ ($\lambda > 0$) de tal forma que

$$\|d\| = \left\| (B(x_k) + \lambda I)^{-1} \nabla f(x_k) \right\| = \Delta_k$$

que é uma equação não linear em λ. Para tornar esta estratégia num procedimento viável, o mecanismo para o cálculo de d deve ser mais expedito do que a solução exacta de (3.16) (veja-se o trabalho de Dennis e Schnabel [27]). Na prática, o objectivo reduz-se a encontrar soluções aproximadas para (3.16), que originam uma redução no modelo m_k, pelo menos maior do que a redução obtida pelo passo de Cauchy. Existem algumas estratégias de cálculo de aproximações à solução de (3.16) das quais se salientam as seguintes: o método pata de cão, apropriado a matrizes $B(x_k)$ definidas positivas e a minimização num subespaço bidimensional aplicada a matrizes $B(x_k)$ indefinidas.

Da mesma forma que na procura unidimensional não é necessário encontrar o comprimento do passo óptimo para se obter convergência global, também nos métodos de região de confiança isso acontece. Embora em termos ideais o objectivo seja encontrar a solução, para efeitos de convergência global, basta encontrar uma aproximação d, dentro da região de confiança, que origine uma redução significativa no modelo m_k. Esta redução significativa pode ser quantificada em termos do passo de Cauchy, que é o minimizante de m_k ao longo da direcção de descida máxima $-\nabla f(x_k)$ na região de confiança Δ_k.

Existem duas relaxações de (3.16), sem que as propriedades teóricas do método fiquem enfraquecidas, que o tornam mais fácil de resolver: a restrição pode ser quase satisfeita e o modelo quadrático minimizado aproximadamente. Surgiram duas classes de métodos eficientes para resolver estas duas possibilidades de relaxação. Uma das classes inclui os métodos aproximados de passo óptimo em que a restrição é relaxada e o valor de λ é aproximado por um processo iterativo. Detalhes deste procedimento podem ser consultados em [27] e [28]. A outra classe define os métodos tipo pata de cão nos quais a minimização do modelo quadrático é relaxada (veja-se por exemplo [28], [67] e [98]).

A estratégia de definição do raio da região de confiança Δ baseia-se na relação entre o modelo m_k e o valor da função objectivo f no iterando x_k. Para um deslocamento d_k define-se a razão

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d_k)}$$

em que o numerador representa a redução actual e o denominador a redução prevista pelo modelo. Como d_k é obtido pela minimização do modelo m_k na região que inclui o deslocamento nulo ($d = 0$), a redução prevista é sempre um valor não negativo. Assim, se ρ_k for negativo, $f(x_k + d_k)$ é maior do que $f(x_k)$ e d_k terá de ser rejeitado.

Se ρ_k for um valor próximo de 1 existe uma concordância, para o deslocamento d_k, entre o modelo m_k e a função f, sendo a região de confiança expandida para a próxima iteração. Se ρ_k for positivo embora afastado de 1, a região de confiança mantém-se. No entanto se este valor estiver próximo de zero, ou no caso em que for negativo, a região de confiança deverá ser contraída.

Um factor importante para um algoritmo baseado em regiões de confiança é a escolha do raio da região de confiança inicial. Se a região de confiança é demasiado grande no contexto da concordância entre o modelo e a função objectivo, o minimizante aproximado do modelo é um fraco indicador do progresso do algoritmo pois pode não corresponder a um redução efectiva da função objectivo. Por outro lado, uma região de confiança muito pequena pode levar a um progresso muito lento de estimação da solução (aconselha-se a consulta de [96]).

Existem três pontos importantes na estratégia de região de confiança - a escolha do raio da região de confiança inicial, o esquema de actualização da região de confiança e o método de resolução de (3.16).

3.4 Procura unidimensional *versus* regiões de confiança

Não parece existir nenhum trabalho que demonstre que uma das abordagens é superior à outra. Ambas têm sido utilizadas ao longo dos tempos nos mais diversos algoritmos e, no futuro, espera-se que as duas técnicas continuem a desempenhar um papel importante nos métodos de optimização.

Alguns investigadores preferem usar a procura unidimensional juntamente com métodos Quasi-Newton baseados em actualizações BFGS porque a matriz pode conservar-se definida positiva, enquanto que com regiões de confiança isso não é garantido. Outros autores alegam que a técnica da região de confiança lida naturalmente com matrizes indefinidas, garantindo que os pontos sela e os máximos são evitados. O trabalho [107] fornece uma comparação entre as duas estratégias.

A procura unidimensional origina códigos de programação mais simples mas, muitas vezes não clarifica o tratamento das matrizes singulares e indefinidas e de outros factores que podem tornar a direcção de procura inaceitável. Os métodos de regiões de confiança apresentam solução para estes problemas, mas requerem em geral algum custo algébrico adicional.

Embora não pareçam existir diferenças significativas no desempenho destas duas técnicas de globalização, no caso do método de Gauss-Newton para mínimos quadrados não lineares isso não acontece. Neste caso, os algoritmos de regiões de confiança, que podem ser vistos como uma combinação entre a direcção Gauss-Newton e a direcção de descida máxima, parecem ser mais robustos e eficientes que os métodos de procura unidimensional que apenas usam a direcção Gauss-Newton. A referência [66] é importante neste domínio.

3.5 Comentários e motivação

A maior parte das propostas que têm surgido na literatura para modificar as equações do método de Newton definem variantes localmente convergentes. Veja-se, por exemplo, as referências [28], [66], [85], [87], [88] e [94]. Também se incluem neste comentário alguns métodos do tipo Quasi-Newton ([6], [26] e [28]). Sendo assim, as técnicas de globalização são uma componente imprescindível no desenvolvimento dos respectivos algoritmos.

A técnica de procura unidimensional baseada numa condição de aceitação de monotonia da função é antiga e tem sido amplamente divulgada. Se forem permitidos acréscimos do valor da função ao longo de um conjunto pré-estabelecido de iterações, conjugados com um controlo *a priori* da grandeza do vector direcção, estratégia conhecida por não monótona com estabilização, surgem vantagens relativamente à estratégia monótona, nomeadamente no que concerne ao número reduzido de cálculos da função e à prevenção do efeito de Maratos.

A implementação de um método do tipo Newton que garante direcções descendentes ao longo do processo iterativo foi um argumento de peso na escolha da estratégia de globalização baseada numa procura unidimensional para este trabalho.

Capítulo 4

Optimização não linear com restrições

No Capítulo 2 falou-se da programação não linear sem restrições. Neste capítulo, vai ser feita uma abordagem à optimização de funções de variável real com restrições em que a função objectivo e/ou as funções de restrição são não lineares nas variáveis.

São apresentadas as técnicas mais conhecidas para a resolução de um problema de optimização não linear com restrições de igualdade: a técnica da programação quadrática sequencial e as técnicas de penalidade sequencial e exacta. Descreve-se com algum detalhe a estratégia escolhida para este trabalho e que é baseada numa técnica de penalidade exacta com uma função Lagrangeana aumentada. Por fim faz-se referência a uma técnica muito recente mas já com resultados promissores, o método dos filtros.

4.1 Introdução

Considere-se o problema com a seguinte formulação geral :

$$\begin{array}{lll} min & f(x) & \\ x \in \mathbb{R}^n & & \\ \text{sujeito a} & c_i(x) = 0 & i = 1, \ldots, m \\ & \widehat{c}_i(x) \geq 0 & i = 1, \ldots, t \\ & l_i \leq x_i \leq u_i & i = 1, \ldots, n. \end{array} \tag{4.1}$$

A função $f(x)$ é designada por função objectivo, $c_i(x)$ $(i = 1, \ldots, m)$ e $\widehat{c}_i(x)$ $(i = 1, \ldots, t)$ são as funções de restrição. Se só existirem restrições do tipo igualdade, $c_i(x) = 0, i = 1, \ldots, m$, o problema é designado por problema com restrições de igualdade. Por sua

vez, se apenas existirem restrições do tipo $\widehat{c}_i(x) \geq 0, i = 1, \ldots, t$, então o problema é caracterizado como tendo restrições do tipo desigualdade. As restrições $l_i \leq x_i \leq u_i$ $(i = 1, \ldots, n)$ são conhecidas por limites simples nas variáveis.

Definição 4.1 *Seja* $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$. *Um ponto* $\overline{x} \in D$ *diz-se admissível se* $c_i(\overline{x}) = 0$ *para* $i = 1, \ldots, m$, $\widehat{c}_i(\overline{x}) \geq 0$ *para* $i = 1, \ldots, t$, *e* $l_i \leq \overline{x}_i \leq u_i$ *para* $i = 1, \ldots, n$.

Um ponto $x \in \mathbb{R}^n$ é admissível se satisfaz todas as restrições.

Neste trabalho considera-se apenas o problema com restrições do tipo igualdade pelo que a formulação (de problema de minimização) fica reduzida à forma:

$$\begin{array}{ll} min & f(x) \\ x \in \mathbb{R}^n & \\ \text{sujeito a} & c_i(x) = 0 \quad i = 1, \ldots, m. \end{array} \qquad (P_{c/rest})$$

A partir deste momento e até ao fim deste trabalho, será esta a formulação utilizada.

Este estudo não se torna limitativo uma vez que qualquer tipo de restrição de desigualdade ou de limites simples, e até mesmo as restrições dos problemas de programação inteira com $x_i \in \{0, 1\}$, podem ser transformadas em restrições de igualdade.

As funções $f(x)$ e $c_i(x)$, $i = 1, \ldots, m$, assumem-se não lineares e duas vezes continuamente diferenciáveis num certo conjunto.

Seja $L(x, \lambda) = f(x) - c(x)^T\lambda$ a função Lagrangeana associada ao problema $(P_{c/rest})$, em que $\lambda \in \mathbb{R}^m$ é o vector dos multiplicadores de Lagrange. As condições de optimalidade para o problema estão enunciadas nas seguintes proposições.

Proposição 4.1 (Condições de optimalidade de primeira ordem) *O ponto* x^* *é mínimo local de* $(P_{c/rest})$ *se existe* $\lambda^* \in \mathbb{R}^m$ *tal que*

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) - \nabla c(x^*)\lambda^* = 0 \\ \nabla_\lambda L(x^*, \lambda^*) &= c(x^*) = 0, \end{aligned}$$

em que $\nabla f(x)$ *é o vector gradiente de* f *e a matriz* $\nabla c(x) \in \mathbb{R}^{n \times m}$ *contém, ao longo das colunas, os vectores gradientes das restrições.*

Um ponto x^* que verifica estas condições diz-se ponto estacionário da Lagrangeana e o par (x^*, λ^*) que verifica a Proposição 4.1 chama-se par Kuhn-Tucker.

Proposição 4.2 (Condições de optimalidade de segunda ordem) *A condição necessária (suficiente) de segunda ordem para que x^*, do par Kuhn-Tucker, seja um mínimo local é que a matriz*

$$\nabla^2_{xx} L(x^*, \lambda^*) = \nabla^2 f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla^2 c_i(x^*)$$

seja semidefinida positiva (definida positiva) no espaço nulo da matriz $\nabla c(x^*)^T$, em que $\nabla^2 f(x^*)$ e $\nabla^2 c_i(x^*), i = 1, \ldots, m$ são as matrizes Hessianas respectivamente da função objectivo e das restrições.

4.2 Função mérito

Na optimização (minimização) com restrições existem dois objectivos a atingir, muitas vezes conflituosos. Por um lado pretende-se reduzir o valor da função objectivo, ao longo do processo iterativo, e por outro satisfazer as restrições. Sempre que é gerado um passo que provoca uma redução substancial na função objectivo, mas que afasta o processo iterativo da região admissível, relativamente à iteração corrente, surge a questão de saber se este passo deverá ou não ser aceite. Não é fácil responder a esta questão. Para avaliar o balanceamento destes dois objectivos surge a função mérito. Esta tem como finalidade, quantificar o compromisso entre os dois objectivos e controlar o algoritmo de forma a que um passo só seja aceite se originar uma redução significativa na função mérito.

Na optimização sem restrições e nos métodos de pontos interiores (pontos sempre admissíveis), a função mérito pode ser a própria função objectivo. Por outro lado, nos métodos de pontos exteriores, isto é, nos que permitem a violação das restrições, é necessário existir uma forma de estabelecer a qualidade do passo de procura.

Uma função mérito muito usada para a generalidade dos problemas de programação não linear é a função l_1

$$\Phi_1(x, r) = f(x) + r \sum_{i=1}^{m} |c_i(x)| \tag{4.2}$$

em que $r > 0$ é o parâmetro de penalidade, que determina o peso que se atribui à satisfação das restrições relativamente à minimização da função objectivo $f(x)$. Esta função mérito é exacta porque para um certo conjunto de valores de r, o minimizante local de $\Phi_1(x, r)$ é a solução do problema $(P_{c/rest})$ ([2]). É importante salientar que $\Phi_1(x, r)$ não é diferenciável.

Outra função mérito é a função Lagrangeana aumentada de Fletcher ([42])

$$\Phi_F(x,r) = f(x) - \lambda(x)^T c(x) + \frac{1}{2} r \sum_{i=1}^{m} c_i(x)^2 \tag{4.3}$$

em que

$$\lambda(x) = \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \nabla c(x)^T \nabla f(x)$$

é uma função diferenciável dos mínimos quadrados para a estimação dos multiplicadores de Lagrange. Esta função mérito para além de ser diferenciável, é também exacta. Mais à frente voltar-se-á a falar desta função.

4.3 Programação quadrática sequencial

A programação quadrática sequencial, do inglês *sequential quadratic programming*, é uma abordagem possível para a resolução de $(P_{c/rest})$ que consiste em aproximá-lo localmente por um problema mais simples, cujas propriedades são bastante conhecidas. É um dos métodos mais eficientes e divulgados para a optimização não linear com restrições gerando direcções de procura através da resolução de uma sequência de subproblemas quadráticos. É apropriado para problemas de qualquer dimensão e pode ser usado num contexto de procura unidimensional ou de regiões de confiança para a sua globalização.

Assim, a função objectivo, $f(x)$, é aproximada por uma forma quadrática, $q(x)$, e as restrições, $c(x)$ por formas lineares, $l(x)$.

Considere-se a expansão de $f(x)$ em série de Taylor no ponto $\overline{x}$

$$f(\overline{x}+d) = f(\overline{x}) + \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2 f(\overline{x}) d + \cdots .$$

Os primeiros três termos da série definem uma forma quadrática em d

$$q(\overline{x}+d) \equiv q(d) = f(\overline{x}) + \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2 f(\overline{x}) d.$$

Da mesma forma, expandindo $c(x)$ em série de Taylor no ponto $\overline{x}$

$$c(\overline{x}+d) = c(\overline{x}) + \nabla c(\overline{x})^T d + \cdots$$

encontra-se uma forma linear em d

$$l(\overline{x}+d) \equiv l(d) = c(\overline{x}) + \nabla c(\overline{x})^T d.$$

O problema $(P_{c/rest})$ vai ser aproximado localmente por um problema mais simples, de programação quadrática:

$$\begin{array}{ll} min & q(d) \\ d \in \mathbb{R}^n & \\ \text{sujeito a} & l_i(d) = 0, \quad i = 1, \ldots, m \end{array}$$

que substituindo origina

$$\begin{array}{ll} min & \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2 f(\overline{x}) d \\ d \in \mathbb{R}^n & \\ \text{sujeito a} & \nabla c_i(\overline{x})^T d = -c_i(\overline{x}) \quad i = 1, \ldots, m. \end{array} \qquad (P_q)$$

Como esta aproximação é válida apenas numa pequena vizinhança de $\overline{x}$, a solução de (P_q), d, não gera necessariamente a solução do problema original $(P_{c/rest})$. Assim, $\overline{x} + d$ funciona como uma nova aproximação à solução de $(P_{c/rest})$, pelo que o processo terá de ser repetido sequencialmente até se obter uma aproximação suficientemente perto de x^* (solução de $(P_{c/rest})$). Esta abordagem define o método de programação quadrática sequencial.

Considere-se a função Lagrangeana do problema (P_q)

$$L_q(d, \eta) = \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2 f(\overline{x}) d - \left(\nabla c(\overline{x})^T d + c(\overline{x})\right)^T \eta \qquad (4.4)$$

em que $\eta \in \mathbb{R}^m$ é o vector dos multiplicadores de Lagrange associado às restrições de (P_q). Calculando os pontos estacionários de L_q em ordem a d e η tem-se

$$\nabla_d L_q(d, \eta) = \nabla f(\overline{x}) + \nabla^2 f(\overline{x}) d - \nabla c(\overline{x}) \eta = 0$$

e

$$\nabla_\eta L_q(d, \eta) = \nabla c(\overline{x})^T d + c(\overline{x}) = 0.$$

Além disso, a função Lagrangeana do problema original $(P_{c/rest})$ é

$$L(x, \lambda) = f(x) - c(x)^T \lambda$$

e a aproximação quadrática obtida a partir da expansão em série de Taylor de $L(x, \lambda)$ em volta de $\overline{x}$ é

$$\begin{aligned} L^Q(\overline{x} + d, \lambda) \equiv L^Q(d, \lambda) &= f(\overline{x}) - c(\overline{x})^T \lambda + \left(\nabla f(\overline{x}) - \nabla c(\overline{x}) \lambda\right)^T d + \\ &+ \frac{1}{2} d^T \left(\nabla^2 f(\overline{x}) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(\overline{x}) \right) d. \end{aligned} \qquad (4.5)$$

Comparando (4.4) com (4.5), verifica-se que, à parte do termo constante $f(\overline{x})$, que existe em $L^Q(d,\lambda)$ e não existe em $L_q(d,\eta)$, o vector dos multiplicadores λ de $(P_{c/rest})$ pode ser obtido a partir de η, vector dos multiplicadores de (P_q). Mais ainda, a matriz do termo quadrático em d, em $L^Q(d,\lambda)$, é $\nabla^2 f(\overline{x}) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(\overline{x}) = \nabla^2_{xx} L(\overline{x},\lambda)$ e, em $L_q(d,\eta)$, é $\nabla^2 f(\overline{x})$. Como é possível provar convergência local quadrática, para o método da programação quadrática sequencial, quando $\nabla^2 f(\overline{x})$ é substituída por $\nabla^2 f(\overline{x}) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(\overline{x})$, ou seja $\nabla^2_{xx} L(\overline{x},\lambda)$, o problema (P_q) deve ser preferencialmente definido da seguinte forma:

$$\begin{array}{ll} min & \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2_{xx} L(\overline{x},\lambda) d \\ d \in \mathbb{R}^n & \\ \text{sujeito a} & \nabla c_i(\overline{x})^T d = -c_i(\overline{x}) \quad i = 1,\ldots,m. \end{array} \qquad (\mathrm{P}_{\overline{q}})$$

A solução do problema de programação quadrática $(\mathrm{P}_{\overline{q}})$ pode ser obtida recorrendo às condições de primeira ordem relativas a $(\mathrm{P}_{\overline{q}})$, que têm por base a função Lagrangeana associada ao problema e que é dada por

$$L_{\overline{q}}(d,\eta) = \nabla f(\overline{x})^T d + \frac{1}{2} d^T \nabla^2_{xx} L(\overline{x},\lambda) d - \left(\nabla c(\overline{x})^T d + c(\overline{x})\right)^T \eta,$$

com $\eta \in \mathbb{R}^m$. Assim , de

$$\begin{cases} \nabla_d L_{\overline{q}}(d,\eta) = 0 \\ \nabla_\eta L_{\overline{q}}(d,\eta) = 0 \end{cases}$$

tem-se

$$\begin{cases} \nabla f(\overline{x}) + \nabla^2_{xx} L(\overline{x},\lambda) d - \nabla c(\overline{x}) \eta = 0 \\ \nabla c(\overline{x})^T d + c(\overline{x}) = 0 \end{cases}$$

que é um sistema linear nas incógnitas d e η. É importante salientar que η, vector dos multiplicadores de $(\mathrm{P}_{\overline{q}})$ é usado para calcular λ do problema original $(P_{c/rest})$.

O método da programação quadrática sequencial, baseado em $\nabla^2_{xx} L(x,\lambda)$, como matriz Hessiana da função quadrática q, é equivalente ao método Lagrange -Newton, no sentido de que o sistema resultante da aplicação do método de Newton à resolução do sistema não linear das condições de optimalidade do problema $(P_{c/rest})$, é idêntico.

Ainda no âmbito da programação quadrática sequencial, é de salientar a versão que, em vez de considerar a Hessiana total da função Lagrangeana, considera uma matriz de dimensão menor, que se obtém através de projecções desta. Esta nova matriz reduzida,

é sob certas condições *standard,* definida positiva na vizinhança da solução. Os métodos que implementam este tipo de estratégia têm confirmado serem eficazes em muitas áreas de aplicação tal como o controlo óptimo. São vantajosos quando a quantidade $(n - m)$ é pequena. Segundo [87, pp 549], nestes métodos a Hessiana reduzida tem maior probabilidade de ser definida positiva do que a Hessiana total, mesmo quando a iteração está longe da solução.

4.4 Técnicas de penalidade

Uma outra estratégia possível para a resolução do problema $(P_{c/rest})$ consiste em transformá-lo num problema sem restrições, de tal forma que a solução deste problema é igual ou está relacionada de uma certa forma com a solução x^* de $(P_{c/rest})$.

O problema original pode ser resolvido pela formulação de uma sequência de subproblemas sem restrições, em que a função objectivo é uma função de penalidade, cuja solução converge para x^* se a função incluir um termo que garante a admissibilidade no limite.

Uma função de penalidade combina a função $f(x)$ (ou $L(x, \lambda)$), com as restrições, $c(x)$, de tal forma que permite que a função $f(x)$ (ou $L(x, \lambda)$) seja minimizada, controlando a violação das restrições, ou seja, penalizando-a. Esta função depende ainda de um ou mais parâmetros que determinam a importância relativa das restrições na função de penalidade. Um incremento gradual da penalidade é imposto ao longo da sequência, forçando os iterandos na direcção da região admissível. Este aumento da penalidade pode originar mau condicionamento da matriz Hessiana da função de penalidade, dificultando a resolução do subproblema sem restrições. Uma vantagem dos métodos de penalidade consiste no facto dos iterandos não terem obrigatoriamente de pertencer à região admissível, sendo portanto adequados aos problemas com restrições do tipo igualdade.

4.4.1 Função de penalidade sequencial l_2

O método de penalidade consiste em transformar o problema $(P_{c/rest})$ num problema sem restrições cuja função objectivo depende de $f(x)$ e das restrições. Considere-se a função de penalidade l_2 representada por

$$Q(x, r) = f(x) + \frac{1}{2} r c(x)^T c(x) \tag{4.6}$$

em que r é o parâmetro de penalidade $(r > 0)$ e o termo de penalidade $\frac{1}{2} r c(x)^T c(x)$ é continuamente diferenciável.

Pode-se mostrar[1] que o minimizante de $Q(x,r)$ designado por $x^*(r)$ verifica

$$\lim_{r\to\infty} x^*(r) = x^*,$$

ou seja, tende para o minimizante de $(P_{c/rest})$ à medida que r aumenta.

Esta técnica, conhecida por técnica de penalidade sequencial, tem o inconveniente de exigir a resolução de um subproblema $\min_{x \in \mathbb{R}^n} Q(x,r)$ para cada valor de r de uma sucessão infinita. A sucessão $\{x^*(r)\}$ de soluções dos subproblemas converge para x^*, ou seja, $x^*(r)$ nunca coincide com x^* para valores finitos de r. Além disso, o facto de $r \to \infty$ pode dar origem a singularidades na matriz Hessiana de $Q(x,r)$ no limite, tornando o problema mal condicionado e a convergência muito lenta, ou mesmo impossibilitando a convergência para o mínimo pretendido. Os livros [41] e [73] são elementos importantes de consulta no âmbito da técnica de penalidade sequencial.

Existem, no entanto, outras funções de penalidade que não sofrem deste inconveniente.

4.4.2 Função de penalidade exacta

No subproblema de minimização sem restrições baseado numa função de penalidade exacta, a solução $x^*(r)$ coincide com x^*, solução de $(P_{c/rest})$, para valores positivos de r tais que $r > \overline{r}$ ([2]). Para determinada escolha do parâmetro r uma única minimização pode ser suficiente para calcular a solução exacta x^* de $(P_{c/rest})$.

O mau condicionamento verificado no subproblema com funções de penalidade sequencial é evitado, uma vez que não é necessário exigir que $r \to \infty$. No entanto, outras dificuldades podem surgir nas funções do tipo penalidade exacta, pois o valor $\overline{r}$ depende de quantidades calculadas em x^*, que sendo desconhecido, obriga a que r tenha de ser estimado e eventualmente ajustado. Podem surgir outros problemas na implementação se forem escolhidos valores inadequados para r. De facto, se r é demasiado pequeno a função de penalidade pode ser ilimitada inferiormente ou a região de convergência para x^* ser muito pequena. Por outro lado se r for grande o problema pode tornar-se mal condicionado.

A diferença principal entre funções de penalidade sequencial e funções de penalidade exacta, reside no facto de na primeira, o problema original $(P_{c/rest})$ ser transformado numa sequência infinita de subproblemas sem restrições, enquanto que na segunda essa sequência é finita. Por esta razão a técnica é designada por penalidade exacta, embora a selecção de um valor adequado para r possa obrigar à resolução de mais do que um subproblema sem restrições.

[1] A sua demonstração pode ser consultada em ([2]).

Quando se definiu função mérito, esta surgiu no contexto de avaliação da qualidade do passo de procura. Na realidade, função mérito e função de penalidade podem ser vistas como sinónimos. Na secção 4.2 foram apresentadas duas funções de penalidade exacta em (4.2) e (4.3). A formulação (4.2), embora o seu cálculo em termos computacionais seja pouco exigente, apenas depende da função objectivo e das restrições, tem um inconveniente que reside no facto das derivadas do termo de penalidade serem descontínuas quando $c(x) = 0$. Para além disso é muito sensível ao efeito de Maratos, podendo levar à rejeição de passos muito proveitosos para o cálculo da solução ([87, pp 436]). O trabalho [8] tece algumas considerações interessantes sobre a técnica de penalidade exacta.

4.5 Função Lagrangeana aumentada

Um outro tipo de função de penalidade exacta é caracterizado por funções Lagrangeanas aumentadas que se obtêm a partir da função Lagrangeana adicionando termos de penalidade (veja-se por exemplo em [36]). Hoje em dia são já conhecidas outras funções aumentadas que se constroem a partir das funções de penalidade mais conhecidas, adicionando termos em função do vector dos multiplicadores. Um exemplo pode ser consultado em [49]. Nos métodos baseados em funções Lagrangeanas aumentadas, define-se uma função que se obtém a partir da função Lagrangeana adicionando-lhe um termo de penalidade, de tal forma que os pontos estacionários da Lagrangeana são também pontos que anulam a primeira derivada da função Lagrangeana aumentada. O termo de penalidade pode ser o da função l_2, em (4.6). A inclusão do termo com os multiplicadores de Lagrange origina uma melhoria substancial do desempenho quando comparada com a função de penalidade l_2. Apresentam ainda a vantagem de não serem afectadas pelo efeito de Maratos.

Considere-se a função Lagrangeana

$$L(x, \lambda) = f(x) - c(x)^T \lambda. \tag{4.7}$$

Uma das condições suficientes para que x^* seja ponto estacionário de $L(x, \lambda)$ é

$$\nabla_x L(x^*, \lambda^*) = 0 \Leftrightarrow \nabla f(x^*) = \nabla c(x^*) \lambda^* \tag{4.8}$$

O ponto x^* é um ponto estacionário de (4.7) quando $\lambda = \lambda^*$ e não é necessariamente um mínimo da função Lagrangeana. Assim, a função (4.7) não é adequada para definir a função objectivo do subproblema sem restrições. À Lagrangeana deve adicionar-se um termo que origine uma função objectivo com matriz das segundas derivadas definida positiva em (x^*, λ^*).

Segundo Fletcher em [42], o termo de penalidade a adicionar à função Lagrangeana deve ter derivada nula em x^*, ou seja, deve ter os mesmos pontos estacionários da Lagrangeana e a sua curvatura deve ser positiva nas direcções normais e nula nas restantes direcções. Estas funções de penalidade são conhecidas por funções Lagrangeanas aumentadas.

O termo $\frac{1}{2}rc(x)^T S(x)c(x)$ tem estas características se $S(x)$ for uma matriz definida positiva e r suficientemente grande. Exemplos de matrizes $S(x)$ são

$$\nabla c(x)^+ \nabla c(x)^{+T}, \nabla c(x)^+ \nabla^2 f(x) \nabla c(x)^{+T}$$

e

$$\nabla c(x)^+ \left[\nabla^2 f(x) + qI\right] \nabla c(x)^{+T} (q > 0)$$

com $\nabla c(x)^+ = \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \nabla c(x)^T$ (veja-se em [42]). A função Lagrangeana aumentada mais simples com parâmetro de penalidade r ($r > 0$) e termo de penalidade $\frac{1}{2}rc(x)^T c(x)$ é

$$\overline{\Phi}(x, \lambda, r) = f(x) - c(x)^T \lambda + \frac{1}{2} rc(x)^T c(x) \tag{4.9}$$

em que $S(x) = I$.

Assim para um certo valor de r, o problema de minimização com restrições ($P_{c/rest}$) pode ser resolvido através do cálculo de

$$\begin{array}{ll} min & \overline{\Phi}(x, \lambda, r) \\ x \in \mathbb{R}^n & \\ \lambda \in \mathbb{R}^m & \end{array} \tag{4.10}$$

Algumas considerações a fazer relativamente à função Lagrangeana aumentada relacionam-se com a estimação dos multiplicadores de Lagrange e com a selecção do parâmetro de penalidade.

Se o subproblema (4.10) for resolvido para um dado r, de modo a conseguir-se calcular duas sucessões $\{x_k(r)\}$ e $\{\lambda_k(r)\}$, de aproximações a x^* e λ^*, o procedimento é caracterizado pelo aumento da dimensão do problema, $n + m$ variáveis, e designa-se por problema primal-dual. No entanto, se for possível encontrar uma relação entre λ e x, de tal forma que, para um dado r, e para uma aproximação $x_k(r)$ de $x^*(r)$, o vector dos multiplicadores é aproximado através da relação $\lambda_k = \lambda(x_k(r))$, o subproblema formula-se da seguinte forma:

$$\begin{array}{ll} min & \overline{\Phi}(x, \lambda, r). \\ x \in \mathbb{R}^n & \end{array} \tag{4.11}$$

O número de variáveis do problema mantém-se e o procedimento é conhecido por método dos multiplicadores e é um problema primal.

Os procedimentos baseados em (4.10) e (4.11) caracterizam, no entanto, técnicas de penalidade sequencial. Se uma função continuamente diferenciável que represente λ em função de x, designada por $\lambda(x)$, for substituir λ em (4.9) obtém-se uma função Lagrangeana aumentada com a forma

$$\Phi(x,r) = f(x) - c(x)^T\lambda(x) + \frac{1}{2}rc(x)^Tc(x) \tag{4.12}$$

que depende apenas da variável x. O subproblema a resolver, para um dado r, reduz-se a

$$\begin{array}{ll} min & \Phi(x,r) \\ x \in \mathbb{R}^n & \end{array} \tag{4.13}$$

e gera um algoritmo de penalidade exacta.

Facchinei e Lucidi em [36] introduzem uma classe de funções de penalidade exacta do tipo Lagrangeana aumentada, com a forma

$$\overline{\overline{\Phi}}(x,r) = f(x) - c(x)^T\lambda(x) + r\frac{1}{a(x)}c(x)^Tc(x) \tag{4.14}$$

em que $a(x)$ é uma função continuamente diferenciável de tal modo que $0 \leq a(x) \leq \overline{a}$ e $\lambda(x)$ é uma função dos multiplicadores continuamente diferenciável. Fazendo em (4.14), $a(x) = 2$ e $\lambda(x) = \left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x)$, $\overline{\overline{\Phi}}(x,r)$ coincide com a função de penalidade exacta proposta por [42] e é a que vai ser usada neste trabalho. No entanto, para $a(x) = \alpha - c(x)^Tc(x)$, com $\alpha > 0$ e para a mesma função $\lambda(x)$, acima indicada, obtém-se a função de penalidade proposta por Di Pillo e Grippo em [92] e [93]. Por sua vez, e para a mesma função $a(x)$, se $\lambda(x) = (\nabla c(x)^T\nabla c(x) + \|c(x)\|^2 I)^{-1}\nabla c(x)^T\nabla f(x)$ consegue-se a função descrita em [69].

Um excelente trabalho de revisão [5], cuja consulta se aconselha, sobre métodos de penalidade sequencial, exacta e sobre o método dos multiplicadores, que cobre o período de 1968 a 1993, foi publicado em 1995 por Boukari e Fiacco e contém trezentas e cinco referências neste domínio.

A escolha do parâmetro de penalidade é um processo complicado tendo sido alvo de numerosos estudos. Existem várias sugestões para o cálculo de r (por exemplo em [37] e [63]) sendo apresentadas mais à frente algumas considerações sobre o assunto.

Os multiplicadores de Lagrange têm um papel muito importante nos métodos de resolução baseados nas funções Lagrangeanas aumentadas. Pode mesmo afirmar-se que a razão de convergência destes métodos depende directamente da precisão da sua estimação

(veja-se como exemplos, os trabalhos [53], [54] e [101]); o ponto x^* é um ponto estacionário de (4.9) se e só se $\lambda = \lambda^*$, ou seja a precisão de λ é fundamental para obtenção do óptimo pretendido.

No capítulo seguinte vai descrever-se a técnica usada neste trabalho para resolver $(P_{c/rest})$ - técnica de penalidade exacta, utilizando uma função Lagrangeana aumentada com substituição do vector dos multiplicadores.

4.6 Método dos filtros

Nesta secção vai fazer-se uma breve referência a uma técnica nova de resolução do problema não linear com restrições, que apresenta já resultados bastante entusiasmadores, que por si só justificam a continuação do investimento no seu estudo. O aparecimento do método dos filtros é recente. O primeiro estudo surge em 1997 com o trabalho de Fletcher e Leyffer [45]. A característica inovadora deste método consiste na substituição da função de penalidade ou da função Lagrangeana aumentada, por um esquema baseado no conceito de filtro, apresentando uma filosofia semelhante à da optimização multi-objectivo. Desta forma, os inconvenientes associados à função de penalidade, tais como a escolha do parâmetro de penalidade e a não diferenciabilidade de algumas funções de penalidade, são evitados. Com base no trabalho pioneiro [45], vai ser apresentada, de uma forma básica, a noção de filtro. Assim, para o problema

$$\begin{array}{ll} min & f(x) \\ x \in \mathbb{R}^n & \\ \text{sujeito a} & c_i(x) \leq 0 \quad i = 1, \ldots, m, \end{array}$$

existem dois objectivos em competição: a minimização da função objectivo f e a satisfação das restrições. O segundo objectivo, $c(x) \leq 0$, pode ser definido também como um problema de minimização

$$min\ h(c(x))$$
$$\text{em que } h(c(x)) = \sum_{j=1}^{m} c_j^+(x)$$

e $c_j^+(x) = max(0, c_j)$. A função de penalidade combina $min\ f(x)$ e $min\ h(c(x))$ num único problema de minimização, enquanto que na perspectiva do filtro, este problema é visto sob o ponto de vista dum problema de optimização multi-objectivo, ou seja, com dois objectivos separados. Convém, no entanto, salientar que esta situação é diferente da optimização multi-objectivo, no sentido de que é fundamental encontrar um ponto x

tal que $h(c(x)) = 0$, sendo portanto prioritária a minimização de $h(c(x))$. Apesar disto, recorre-se a um conceito da optimização multi-objectivo, o conceito de dominação.

Definição 4.2 *Diz-se que um par* $(f(x_k), h(c(x_k)))$ *domina outro par* $(f(x_l), h(c(x_l)))$ *se e só se* $f(x_k) < f(x_l)$ *e* $h(c(x_k)) \leq h(c(x_l))$ *ou* $f(x_k) \leq f(x_l)$ *e* $h(c(x_k)) < h(c(x_l))$.

Com esta definição é agora possível definir o conceito de filtro.

Definição 4.3 *Um filtro é uma lista de pares* $(f(x_l), h(c(x_l)))$ *tal que nenhum par domina outro. Um par* $(f(x_k), h(c(x_k)))$ *inclui-se no filtro se não é dominado por nenhum outro par do filtro.*

Cada par no filtro gera um bloco de pares não aceitáveis cuja união representa o conjunto de pares que não são aceites pelo filtro. A ideia chave subjacente ao conceito de filtro é a de que este é usado para aceitar ou rejeitar passos. O critério de descida da função de penalidade é substituído pela exigência do novo par ser aceite no filtro. O filtro permite uma certa não-monotonia quando comparado com a abordagem da função de penalidade, o que na prática pode mostrar-se vantajoso. No trabalho [45], os autores utilizam o conceito de filtro no âmbito da programação quadrática sequencial como método iterativo básico e recorrem à técnica de regiões de confiança para globalização, mas referem que pode ser aplicado noutras circunstâncias. Sugerem ainda algumas eventuais heurísticas para eliminar situações óbvias em que o método pode falhar. Segundo Fletcher e Leyffer, a questão pertinente da convergência global ainda se mantém em aberto e tem suscitado discussões interessantes entre os peritos na área.

Continuando o estudo dos filtros, Fletcher, Leyffer e Toint em [46] apresentam um mecanismo que, quando usado de forma apropriada, prova a convergência global nos métodos tipo filtro. Para isso, consideram uma situação básica na qual é usado um algoritmo de programação linear sequencial com região de confiança e filtro. Uma característica importante da prova é a de que várias das heurísticas utilizadas no trabalho anterior são agora desnecessárias.

São de ressalvar ainda, entre outros, os trabalhos [44] e [103]. Em [44] definem-se duas variantes de algoritmos tipo filtro baseadas na programação quadrática sequencial com regiões de confiança, provando-se que são globalmente convergentes para pontos estacionários. Os autores afirmam que estes algoritmos são muito eficientes na prática e apresentam propriedades razoáveis de convergência. Salientam ainda que o estudo da convergência não está completo, nomeadamente em termos da convergência local e da convergência para pontos que satisfazem as condições de segunda ordem.

Em [103], M. Ulbrich, S. Ulbrich e Vicente usam a técnica dos filtros proposta em [45] para globalizar um algoritmo primal-dual de pontos interiores para a programação não

linear. Este novo algoritmo decompõe o passo primal-dual obtido das condições necessárias de primeira ordem perturbadas num passo normal e num tangencial, cujos tamanhos são controlados por um parâmetro tipo região de confiança. Cada entrada no filtro define um par de coordenadas, uma associada à componente normal do passo e a outra à componente tangencial. Prova-se a convergência global para pontos estacionários.

Recentemente, Benson, Shanno e Vanderbei ([3]) propuseram três abordagens diferentes para implementar um método baseado em filtros no controlo do comprimento do passo num algoritmo de pontos interiores. O trabalho apresenta um estudo computacional exaustivo que inclui a comparação com o algoritmo *standard* LOQO ([104]) baseado numa função mérito.

Ainda no âmbito do método dos filtros, e por iniciativa do Centro de Matemática da Universidade de Coimbra, foi realizado um encontro "First Workshop on Nonlinear Optimization: Interior-Point and Filter Methods" que decorreu na Quinta das Lágrimas em Coimbra, de 27 a 30 de Outubro de 1999. Estiveram presentes, em número significativo, os principais investigadores nesta área, tendo sido apresentadas várias comunicações demonstrando o esforço que se tem vindo a investir nesta recente técnica da optimização.

4.7 Síntese

As três abordagens referidas para a resolução de um problema de optimização não linear com restrições surgem com base em objectivos centrais distintos.

O método da programação quadrática sequencial aproxima localmente o problema original por um problema quadrático do mesmo tipo, originando um processo sequencial até convergir. As técnicas de penalidade têm como objectivo transformar o problema original num outro sem restrições - subproblema. A resolução deste pode ser feita, por exemplo, por um método do tipo Newton, e a solução do problema original obtém-se ao fim de uma sequência infinita ou finita de subproblemas, consoante a penalidade é sequencial ou exacta. Finalmente, o método dos filtros surge da formulação de um problema com dois objectivos separados. O primeiro minimiza a função objectivo e o outro minimiza uma função das restrições, sendo esta prioritária.

Capítulo 5

Método pseudo-Newton modificado

No capítulo anterior foram introduzidas algumas técnicas para a resolução do problema $(P_{c/rest})$. Neste trabalho optou-se pela implementação e análise de uma técnica de penalidade exacta baseada na função Lagrangeana aumentada $\Phi(x,r)$ (em (4.12)), que recorre à substituição do vector dos multiplicadores de Lagrange λ por uma função $\lambda(x)$ (veja-se em (5.1)). Neste capítulo, começa-se por descrever a função para a estimação do vector dos multiplicadores λ, calculada a partir das condições de primeira ordem, apresentando-se ainda a fórmula do seu Jacobiano $\nabla\lambda(x)^T$. Como o parâmetro de penalidade também tem um papel muito importante na convergência do processo iterativo, apresenta-se uma estratégia de actualização que tem dado resultados promissores. Para resolver o correspondente problema sem restrições, com função objectivo $\Phi(x,r)$, usa-se um método do tipo Newton. O gradiente da função Lagrangeana aumentada, $\nabla\Phi$, e a matriz Hessiana, $\nabla^2\Phi$, têm expressões complicadas que envolvem muitos cálculos. No sentido de simplificar estas fórmulas apresentam-se umas aproximações, respectivamente $D\Phi$ e $D^2\Phi$ designadas por pseudo-derivadas. O sistema resultante vai ser designado por sistema pseudo-Newton e a correspondente direcção, por direcção pseudo-Newton. Faz-se o estudo de convergência local do método pseudo-Newton. Mostra-se que o sistema pseudo-Newton surge do sistema do método de Newton com a matriz dos coeficientes modificada. Introduz-se a direcção pseudo-Newton modificada, que consiste na implementação duma estratégia semelhante à que foi apresentada no Capítulo 2. Por fim é feito o estudo da convergência global do método pseudo-Newton modificado.

5.1 Estimação dos multiplicadores de Lagrange

A fórmula utilizada neste trabalho para a estimação do vector dos multiplicadores baseia-se nas condições de primeira ordem. Funciona como uma fórmula de substituição do vector dos multiplicadores em que λ aparece definido como uma função de x e λ^* pode ser estimado em cada iteração.

Neste trabalho, a estratégia utilizada na resolução do problema $(P_{c/rest})$ consiste em utilizar a função Lagrangeana aumentada, $\overline{\Phi}(x, \lambda, r)$, que tem a forma

$$\overline{\Phi}(x, \lambda, r) = f(x) - c(x)^T\lambda + \frac{1}{2}rc(x)^Tc(x).$$

Se $f(x)$ e $c(x)$ forem funções não lineares, a função $\overline{\Phi}$ é também não linear em x e um ponto estacionário de $\overline{\Phi}$ determina-se pela resolução do sistema de equações não lineares

$$\begin{cases} \nabla_x\overline{\Phi}(x, \lambda, r) = 0 \\ \nabla_\lambda\overline{\Phi}(x, \lambda, r) = 0, \end{cases}$$

em ordem a x e λ, e de dimensão $n+m$. Para reduzir a dimensão do problema, é possível estimar o valor de λ^* do par (x^*, λ^*), usando fórmulas de estimação de λ. Estas funções substituídas em $\overline{\Phi}(x, \lambda, r)$ originam uma função Lagrangeana aumentada, função apenas de x, $\overline{\Phi}(x, \lambda(x), r) \equiv \Phi(x, r)$. A dimensão do problema mantém-se em n, pressupondo que r é uma constante e a formulação matemática é agora

$$\begin{array}{ll} min & \Phi(x, r). \\ x \in \mathbb{R}^n & \end{array} \qquad (P_{s/rest})$$

Considere-se a seguinte proposição.

Proposição 5.1 *Sejam $f(x)$ e $c_i(x)$, $i = 1, \ldots, m$, funções continuamente diferenciáveis em $D \subset \mathbb{R}^n$. Suponha ainda que a matriz $\nabla c(x)_{n\times m}$ tem característica m $(m \leq n)$ e que as condições de primeira ordem, no ponto (x^*, λ^*) se verificam:*

$$\nabla_x L(x^*, \lambda^*) = 0 \text{ e } \nabla_\lambda L(x^*, \lambda^*) = 0,$$

em que $L(x, \lambda) = f(x) - c(x)^T\lambda$ é a função Lagrangeana associada ao problema $(P_{c/rest})$ e $\lambda \in \mathbb{R}^m$. Então existe uma função $\lambda(x) : \mathbb{R}^n \to \mathbb{R}^m$, que determina a relação entre λ e o vector x, definida por

$$\lambda(x) = \left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x). \qquad (5.1)$$

Prova. Da condição $\nabla_x L(x,\lambda) = 0$ tira-se que $\nabla c(x)\lambda = \nabla f(x)$. A solução deste sistema de n equações com m incógnitas deve ser calculada pelo método dos mínimos quadrados. Assim, $\nabla c(x)^T \nabla c(x)\lambda = \nabla c(x)^T \nabla f(x)$ ou seja, $\lambda(x) = \nabla c(x)^+ \nabla f(x)$ em que $\nabla c(x)^+$ é a inversa generalizada à esquerda da matriz $\nabla c(x)$, definida por $\nabla c(x)^+ = \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \nabla c(x)^T$. ■

Considere-se o seguinte lema:

Lema 5.1 *Considerem-se as condições da Proposição 5.1. No cálculo da solução do problema ($P_{s/rest}$) considere a função de substituição $\lambda(x)$ definida por (5.1). Então a função $\lambda(x)$ é diferenciável e a matriz $\nabla\lambda(x)$ é dada por*

$$\begin{aligned} \nabla\lambda(x) &= \left[\nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i(x)\nabla^2 c_i(x)\right] \left(\nabla c(x)^+\right)^T + \\ &\quad + \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T\right) \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \end{aligned} \tag{5.2}$$

em que $v(x) = \nabla f(x) - \nabla c(x)\lambda(x)$, e_i é a coluna i da matriz Identidade de ordem m, $\nabla^2 f(x)$ é a matriz Hessiana de $f(x)$ e $\nabla^2 c_i(x)$ é a Hessiana, de ordem n, da função de restrição $c_i(x)$ $(i = 1, \ldots, m)$.

Prova. Repete-se aqui a prova apresentada em [77] e [76]. A equação $\nabla f(x) - \nabla c(x)\lambda(x) = 0$ é equivalente à seguinte conjunção

$$\begin{cases} v(x) &= \nabla f(x) - \nabla c(x)\lambda(x) \\ \nabla c(x)^T v(x) &= 0 \end{cases}$$

para algum vector $v(x) \in \mathbb{R}^n$, desde que a característica de $\nabla c(x)$ seja m. Derivando estas equações em ordem a x obtém-se

$$\begin{cases} \nabla v(x) = \nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i(x)\nabla^2 c_i(x) - \nabla c(x)\nabla\lambda(x)^T \\ \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T\right)^T + \nabla c(x)^T \nabla v(x) = 0 \end{cases}$$

Multiplicando a primeira expressão à esquerda por $\nabla c(x)^T$ e substituindo na segunda chega-se a

$$\begin{aligned} \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T\right)^T &+ \nabla c(x)^T \nabla^2 f(x) - \nabla c(x)^T \sum_{i=1}^{m} \lambda_i(x)\nabla^2 c_i(x) - \\ &- \nabla c(x)^T \nabla c(x)\nabla\lambda(x)^T = 0. \end{aligned}$$

Explicitando em função do último termo desta equação vem

$$\begin{aligned}\nabla c(x)^T \nabla c(x) \nabla \lambda(x)^T &= \nabla c(x)^T \nabla^2 f(x) - \nabla c(x)^T \sum_{i=1}^{m} \lambda_i(x) \nabla^2 c_i(x) + \\ &+ \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T \right)^T\end{aligned}$$

ou, como $\nabla c(x)^T \nabla c(x)$ é uma matriz não singular, tem-se

$$\begin{aligned}\nabla \lambda(x)^T &= \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \nabla c(x)^T \nabla^2 f(x) - \\ &- \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \nabla c(x)^T \sum_{i=1}^{m} \lambda_i(x) \nabla^2 c_i(x) + \\ &+ \left(\nabla c(x)^T \nabla c(x)\right)^{-1} \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T \right)^T.\end{aligned}$$

■

Voltando à função objectivo do problema $(P_{s/rest})$

$$\Phi(x, r) = f(x) - c(x)^T \lambda(x) + \frac{1}{2} r c(x)^T c(x), \tag{5.3}$$

com $\lambda(x) = \nabla c(x)^+ \nabla f(x)$ tem-se então,

Proposição 5.2 *Considerem-se as condições da Proposição 5.1 e a função objectivo do problema $(P_{s/rest})$ apresentada em (5.3). Então $\Phi(x, r)$ é diferenciável, tem primeira derivada, em ordem a x, dada por*

$$\nabla \Phi(x, r) = \nabla f(x) - \nabla c(x) \lambda(x) - \nabla \lambda(x) c(x) + r \nabla c(x) c(x), \tag{5.4}$$

e matriz Hessiana das segundas derivadas definida por

$$\begin{aligned}\nabla^2 \Phi(x, r) &= \nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i(x) \nabla^2 c_i(x) - \nabla c(x) \nabla \lambda(x)^T - \\ &- \left[\nabla \lambda(x) - r \nabla c(x)\right] \nabla c(x)^T - \\ &- \sum_{i=1}^{m} c_i(x) \nabla^2 \lambda_i(x) + r \sum_{i=1}^{m} c_i(x) \nabla^2 c_i(x)\end{aligned} \tag{5.5}$$

em que $\lambda(x)$ e $\nabla \lambda(x)$ foram definidos respectivamente em (5.1) e (5.2), $\nabla^2 f(x)$ é a matriz Hessiana de $f(x)$, $\nabla^2 c_i(x)$ é a Hessiana da restrição $c_i(x)$ $(i = 1, \ldots, m)$ e $\nabla^2 \lambda_i(x)$ é também a Hessiana do elemento $\lambda_i(x)$ $(i = 1, \ldots, m)$ do vector função $\lambda(x)$.

Como o problema $(P_{s/rest})$ se resolve através da determinação dos pontos estacionários de $\Phi(x,r)$, $\nabla\Phi(x,r) = 0$, tem-se pela aplicação do método de Newton à resolução deste sistema não linear de equações,

$$\nabla^2\Phi(x,r)d_N = -\nabla\Phi(x,r) \tag{5.6}$$

em que $\nabla^2\Phi(x,r)$, em (5.5), é a Hessiana de $\Phi(x,r)$ e $\nabla\Phi(x,r)$, em (5.4), é o vector gradiente de $\Phi(x,r)$. O vector solução do sistema linear (5.6), d_N, define a direcção de procura Newton, ao longo da qual se deve procurar o minimizante de $\Phi(x,r)$.

Como o sistema (5.6) vem da aproximação de $\nabla\Phi(x,r) = 0$ por um sistema linear, a que corresponde uma aproximação quadrática a $\Phi(x,r)$, o ponto $x + d_N$ é ponto estacionário do modelo quadrático de Φ e será minimizante, se $\nabla^2\Phi(x,r)$ for uma matriz definida positiva. No entanto não é necessariamente ponto estacionário de $\nabla\Phi(x,r)$, ou minimizante de Φ. Assim o cálculo do vector direcção d_N, deve ser repetido, originando um processo iterativo. A nova aproximação à solução é agora $x + d_N$.

5.2 Parâmetro de penalidade

Como já foi referido, num problema de optimização sem restrições em que a função objectivo é uma função de penalidade resultante da transformação de um problema com restrições num sem restrições, existe outro factor de extrema importância no desempenho, que é o parâmetro de penalidade, r. Nas técnicas de penalidade exacta, o parâmetro r deve ser um valor finito para que a solução do problema sem restrições na variável, $(P_{s/rest})$, coincida com a solução do problema original, com restrições de igualdade, $(P_{c/rest})$. A questão seguinte que se coloca é saber qual o valor mais adequado para esse parâmetro, por forma a que a implementação seja bem sucedida e a convergência seja a mais rápida possível.

Existem, hoje em dia, várias sugestões para o cálculo deste valor e a maior parte delas definem esquemas que ajustam esse valor, de iteração para iteração, no processo iterativo que leva ao cálculo da solução do problema $(P_{s/rest})$. Alguns esquemas conservam o valor do parâmetro ao longo de várias iterações, dependendo da posição da aproximação em relação ao mínimo (veja-se, por exemplo, [9], [68] e o trabalho [23] de Dennis, El-Alem e Maciel no âmbito de algoritmos de regiões de confiança).

Adoptou-se neste trabalho, um esquema já testado anteriormente na resolução de problemas do mesmo tipo, com algum sucesso. Os detalhes da sua implementação, bem como resultados comparativos com outros esquemas podem ser vistos em [37] e [39].

Apresenta-se a seguir, o esquema em procedimento algorítmico que define r como uma função de x, através do valor das funções de restrição, em cada iteração k.

Algoritmo 5.1 (Parâmetro de penalidade)

Se $(\|c(x_k)\|_2 \leq \xi)$ ***Então*** $\{r \leftarrow 1.0\}$
Senão
  Se $\left(c(x_k)^T\lambda(x_k) \geq \dfrac{\|c(x_k)\|_2}{2}\right) \vee (\|c(x_k)\|_2 > 1.0)$ ***Então*** $\{r \leftarrow 10.0\}$
  Senão $\{r \leftarrow \dfrac{1}{\|c(x_k)\|_2}\}$
  FimSe
FimSe

O procedimento baseia-se em três ideias dependendo da proximidade da aproximação em relação à região admissível. Assim, na vizinhança ξ do conjunto admissível não é tão necessário recorrer à penalidade e r é considerado uma constante unitária.

Quando $c(x_k)^T\lambda(x_k)$ toma valores superiores ou iguais a $\dfrac{\|c(x_k)\|_2}{2}$, os termos $-c(x)^T\lambda(x)+\dfrac{1}{2}\|c(x)\|_2$ da função de penalidade dão um contributo negativo para $\Phi(x,r)$, começando o problema a afastar-se da solução pretendida. Nesta situação, ou mesmo quando o ponto se encontra muito afastado da região admissível ($\|c(x_k)\|_2 > 1.0$) a penalidade deve ser reforçada e r toma o valor 10.0.

Para os restantes casos, r é definido por uma função inversamente proporcional à proximidade da região admissível, $\dfrac{1}{\| c(x_k) \|_2}$. Existindo a salvaguarda imposta pela condição

$$\underline{\text{Se}}\,(\| c(x_k) \|_2 \leq \xi\,)\,\underline{\text{Então}}\,\{r \leftarrow 1.0\}$$

nunca se corre o risco de ter r infinito.

5.3 Pseudo-derivadas

Como se pôde ver na secção 5.1, as fórmulas (5.4) e (5.5), respectivamente para o gradiente e Hessiana de $\Phi(x,r)$, envolvem já uma complexidade razoável. Assim, $\nabla\Phi(x,r)$ depende de $\nabla\lambda(x)$ e $\nabla^2\Phi(x,r)$ depende ainda de $\nabla^2\lambda_i(x)$ e de $\nabla^2 c_i(x)$. Se em (5.4) e (5.5) $\lambda(x)$ e $\nabla\lambda(x)$ forem substituídos por (5.1) e (5.2), respectivamente, as expressões para $\nabla\Phi(x,r)$

e $\nabla^2\Phi(x,r)$ apresentam as seguintes formas:

$$\begin{aligned}\nabla\Phi(x,r) &= \nabla f(x) - \nabla c(x)\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x) + r\nabla c(x)c(x) - \\ &\quad - \left\{\left[\nabla^2 f(x) - \sum_{i=1}^{m}\left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x)\right)_i \nabla^2 c_i(x)\right]\right. \\ &\quad \left.\left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\right)^T + \left(\sum_{i=1}^{m}\nabla^2 c_i(x)v(x)e_i^T\right)\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\right\} c(x)\end{aligned} \tag{5.7}$$

e

$$\begin{aligned}\nabla^2\Phi(x,r) &= \nabla^2 f(x) - \sum_{i=1}^{m}\left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x)\right)_i \nabla^2 c_i(x) - \\ &\quad -\nabla c(x)\{\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\left[\nabla^2 f(x) - \right. \\ &\quad \left. - \sum_{i=1}^{m}\left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x)\right)_i \nabla^2 c_i(x)\right] + \\ &\quad + \left(\nabla c(x)^T\nabla c(x)\right)^{-1}\left(\sum_{i=1}^{m}\nabla^2 c_i(x)v(x)e_i^T\right)^T\Big\} - \\ &\quad - \left\{\left[\nabla^2 f(x) - \sum_{i=1}^{m}\left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\nabla f(x)\right)_i \nabla^2 c_i(x)\right]\right. \\ &\quad \left(\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\nabla c(x)^T\right)^T + \\ &\quad \left. + \left(\sum_{i=1}^{m}\nabla^2 c_i(x)v(x)e_i^T\right)\left(\nabla c(x)^T\nabla c(x)\right)^{-1} - r\nabla c(x)\right\}\nabla c(x)^T - \\ &\quad - \sum_{i=1}^{m} c_i(x)\nabla^2\lambda_i(x) + r\sum_{i=1}^{m} c_i(x)\nabla^2 c_i(x).\end{aligned} \tag{5.8}$$

É possível aproximar as expressões do vector gradiente, $\nabla\Phi(x,r)$, e da Hessiana, $\nabla^2\Phi(x,r)$, para tornar a construção do sistema das equações Newton, em (5.6), mais

fácil e menos dispendiosa. Se na expressão para $\nabla\lambda(x)$

$$\begin{aligned}\nabla\lambda(x) &= \left[\nabla^2 f(x) - \sum_{i=1}^{m}\lambda_i(x)\nabla^2 c_i(x)\right](\nabla c(x)^+)^T + \\ &+ \left(\sum_{i=1}^{m}\nabla^2 c_i(x)v(x)e_i^T\right)\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\end{aligned}$$

não forem incluídos os termos que vêm em função das Hessianas das funções de restrição, $\nabla^2 c_i(x)$, obtém-se uma aproximação à matriz $\nabla\lambda(x)$, designada por pseudo-derivada. Considere-se a seguinte definição.

Definição 5.1 *Considerem-se as condições do Lema 5.1. Se as funções de restrição forem aproximadas por funções lineares, localmente, na vizinhança do ponto x, de raio d, de tal forma que*

$$c(x+d) \approx l(d) = c(x) + \nabla c(x)^T d, \tag{5.9}$$

define-se a pseudo-derivada da função $\lambda(x)$ por

$$D\lambda(x) = \nabla^2 f(x)\left(\nabla c(x)^+\right)^T. \tag{5.10}$$

Definição 5.2 *Considerem-se as condições do Lema 5.1. Se as funções de restrição são aproximadas localmente pelas funções lineares (5.9) e $f(x)$ aproximada por uma quadrática, define-se o vector das primeiras pseudo-derivadas de $\Phi(x,r)$ por*

$$\begin{aligned}D\Phi(x,r) &= \nabla f(x) - \nabla c(x)\lambda(x) + r\nabla c(x)c(x) - \\ &\quad -\nabla^2 f(x)\left(\nabla c(x)^+\right)^T c(x)\end{aligned} \tag{5.11}$$

e a matriz das segundas pseudo-derivadas por

$$\begin{aligned}D^2\Phi(x,r) &= \nabla^2 f(x) - \nabla c(x)\nabla c(x)^+\nabla^2 f(x) - \\ &\quad -\nabla^2 f(x)\left(\nabla c(x)^+\right)^T\nabla c(x)^T + r\nabla c(x)\nabla c(x)^T.\end{aligned} \tag{5.12}$$

Nota: O penúltimo termo da expressão de $\nabla^2\Phi(x,r)$, em (5.8), depende das Hessianas de $\lambda_i(x)\,(i=1,\ldots,m)$ e vai tender para zero quando $x \to x^*$. Face às condições enunciadas na Definição 5.1, o único termo que não seria nulo em $\nabla^2\lambda_i(x)$ viria em função das terceiras derivadas de $f(x)$. Assim, ao aproximar localmente $f(x)$ por um modelo quadrático, esse termo também se anula.

Com as aproximações definidas, $D\Phi(x,r)$, em (5.11), e $D^2\Phi(x,r)$, em (5.12), o sistema Newton passa a designar-se por sistema pseudo-Newton e a direcção calculada do sistema por direcção pseudo-Newton. A partir de agora esta direcção vai ser denotada por d_{PN},

$$D^2\Phi(x,r)d_{PN} = -D\Phi(x,r). \tag{5.13}$$

5.4 Sistema pseudo-Newton

Daqui para a frente, e com o intuito de simplificar a notação, não será feita referência ao facto de Φ e respectivas derivadas e aproximações às derivadas, dependerem do parâmetro de penalidade r. Nesta secção vai analisar-se a relação entre as componentes do sistema Newton e do sistema pseudo-Newton e mostrar-se que este último pode ser colocado numa nova formulação em que a única componente diferente do sistema Newton é a matriz dos coeficientes.

Enquanto que no método de Newton o processo iterativo é definido pela equação

$$x_{k+1}^N = x_k - \nabla^2\Phi(x_k)^{-1}\nabla\Phi(x_k),$$

o sistema pseudo-Newton (5.13) origina a equação iterativa

$$\begin{aligned} x_{k+1}^{PN} &= x_k - D^2\Phi(x_k)^{-1}D\Phi(x_k) \\ &= x_k - \left[\nabla^2\Phi(x_k) - \Delta(x_k)\right]^{-1}\left[\nabla\Phi(x_k) - \epsilon(x_k)\right] \end{aligned}$$

em que $\Delta(x) \in \mathbb{R}^{n\times n}$ e $\epsilon(x) \in \mathbb{R}^n$. A matriz $\Delta(x)$ pode ser vista como a perturbação existente em $\nabla^2\Phi(x)$ e $D^2\Phi(x)$ a matriz perturbada. O vector $\epsilon(x)$ pode ser interpretado como a perturbação existente em $\nabla\Phi(x)$ e $D\Phi(x_k)$ o vector perturbado. Estas duas novas entidades são definidas respectivamente por

$$\begin{aligned} \Delta(x) &= \nabla^2\Phi(x) - D^2\Phi(x) = \\ &= -\sum_{i=1}^m \lambda_i(x)\nabla^2 c_i(x) + \\ &\quad +\nabla c(x)\left[\nabla c(x)^+\left(\sum_{i=1}^m \lambda_i(x)\nabla^2 c_i(x)\right) - \left(\nabla c(x)^T\nabla c(x)\right)^{-1}\left(\sum_{i=1}^m \nabla^2 c_i(x)v(x)e_i^T\right)^T\right] + \\ &\quad +\left[\sum_{i=1}^m \lambda_i(x)\nabla^2 c_i(x)\left(\nabla c(x)^+\right)^T - \left(\sum_{i=1}^m \nabla^2 c_i(x)v(x)e_i^T\right)\left(\nabla c(x)^T\nabla c(x)\right)^{-1}\right]\nabla c(x)^T - \\ &\quad -\sum_{i=1}^m c_i(x)\nabla^2\lambda_i(x) + r\sum_{i=1}^m c_i(x)\nabla^2 c_i(x) \end{aligned} \tag{5.14}$$

e

$$\begin{aligned}\epsilon(x) &= \nabla\Phi(x) - D\Phi(x) = (D\lambda(x) - \nabla\lambda(x))\, c(x) = \\ &= \left[\sum_{i=1}^{m} \lambda_i(x)\nabla^2 c_i(x) \left(\nabla c(x)^+\right)^T - \right. \\ &\quad \left. - \left(\sum_{i=1}^{m} \nabla^2 c_i(x) v(x) e_i^T\right) \left(\nabla c(x)^T \nabla c(x)\right)^{-1}\right] c(x). \qquad (5.15)\end{aligned}$$

De (5.15) verifica-se que $\epsilon(x)$ depende de $c(x)$ pelo que se x pertencer à região admissível vem $\epsilon(x) = 0$ e portanto $D\Phi(x) = \nabla\Phi(x)$. De (5.14) tem-se que se x for admissível e para problemas com restrições lineares, obtém-se $\Delta(x) = 0$ e $D^2\Phi(x) = \nabla^2\Phi(x)$. Para x^*, solução do problema $(P_{c/rest})$, verifica-se $\epsilon(x^*) = 0$ mas $\Delta(x^*) \neq 0$, uma vez que $\Delta(x)$ depende de $c(x)$ mas também de $\nabla^2 c_i(x), i = 1, \ldots, m$.

A matriz pseudo-Hessiana, $D^2\Phi(x)$, goza de algumas propriedades que serão seguidamente enunciadas.

Definição 5.3 *Seja $\{\mathbb{R}^n, \|.\|\}$ um espaço linear normado. Um conjunto $L \subset \mathbb{R}^n$ é compacto se e só se qualquer sucessão $\{x_k\}$ em L contém uma subsucessão $\{x_{k_i}\}$ que converge para um limite em L.*

Considerem-se as condições da Proposição 5.1, $\Phi : L \subset D \subset \mathbb{R}^n \to \mathbb{R}$ e as seguintes hipóteses:

Hipótese 5.1 *A inversa da matriz Hessiana de Φ no ponto $x^* \in L$ existe e é limitada em norma*

$$\left\|\nabla^2\Phi^{-1}(x^*)\right\| \leq \beta, \quad (0 < \beta < \infty).$$

Devido à continuidade de $\nabla^2\Phi(x)$ no compacto, $\nabla^2\Phi(x)^{-1}$ existe em L.

Hipótese 5.2 *A matriz $\Delta(x)\nabla^2\Phi(x)^{-1}$ é convergente, isto é,*

$$\left|vp_i(\Delta(x)\nabla^2\Phi(x)^{-1})\right| < 1,$$

para $i = 1, \ldots, n$ e para todo o $x \in L$, em que vp denota valor próprio.

Na sequência desta hipótese, tem-se que $I - \Delta(x)\nabla^2\Phi(x)^{-1}$ é não singular (Lema B.4 no Apêndice B e [72, pp 119]). Além disso,

Propriedade 5.1 *De acordo com as Hipóteses 5.1 e 5.2, a matriz pseudo-Hessiana de* $\Phi(x,r)$, *dada por (5.12), é bem definida no compacto* L.

Nas condições indicadas, como

$$D^2\Phi(x) = \nabla^2\Phi(x) - \Delta(x)$$

com $\nabla^2\Phi(x)$ e $\Delta(x)$ definidas respectivamente por (5.8) e (5.14),

$$D^2\Phi(x) = \left(I - \Delta(x)\nabla^2\Phi(x)^{-1}\right)\nabla^2\Phi(x)$$

ou seja

$$D^2\Phi(x)^{-1} = \nabla^2\Phi(x)^{-1}\left(I - \Delta(x)\nabla^2\Phi(x)^{-1}\right)^{-1} \tag{5.16}$$

e $D^2\Phi(x)$ é bem definida no compacto L.

Propriedade 5.2 *A matriz* $D^2\Phi(x)$, *definida em (5.12), é simétrica para todo o* x.

Considere-se a matriz $D^2\Phi(x)$ com a forma

$$D^2\Phi(x) = \nabla^2 f(x) - \nabla c(x) D\lambda(x)^T - D\lambda(x)\nabla c(x)^T + r\nabla c(x)\nabla c(x)^T.$$

O primeiro e quarto termos são simétricos. O segundo é simétrico do terceiro termo, logo $D^2\Phi$ é uma matriz simétrica pois é a soma de matrizes simétricas.

O sistema, designado pseudo-Newton,

$$D^2\Phi_k d_{PN_k} = -D\Phi_k$$

pode ser posto na forma

$$\mathbf{C}_k^{-1} d_{PN_k} = -\nabla\Phi_k$$

em que a matriz dos coeficientes apresenta a seguinte expressão

$$\mathbf{C}_k^{-1} = \left[\left(I - o_k\epsilon_k\nabla\Phi_k^T\right)^{-1}\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)\nabla^2\Phi_k\right]$$

com $o_k = \dfrac{1}{\|\nabla\Phi_k\|_2^2}$. De facto, a partir de

$$\begin{aligned}
d_{PN_k} &= -D^2\Phi_k^{-1}(\nabla\Phi_k - \epsilon_k) \\
&= -D^2\Phi_k^{-1}\nabla\Phi_k + D^2\Phi_k^{-1}\epsilon_k\left[o_k\,\nabla\Phi_k^T\nabla\Phi_k\right] \\
&= \left(-D^2\Phi_k^{-1} + D^2\Phi_k^{-1}o_k\,\epsilon_k\,\nabla\Phi_k^T\right)\nabla\Phi_k \\
&= \left[-D^2\Phi_k^{-1}(I - o_k\epsilon_k\nabla\Phi_k^T)\right]\nabla\Phi_k \\
&= -\left[(I - o_k\epsilon_k\nabla\Phi_k^T)^{-1}D^2\Phi_k\right]^{-1}\nabla\Phi_k \\
&= -\left[(I - o_k\epsilon_k\nabla\Phi_k^T)^{-1}(I - \Delta_k\nabla^2\Phi_k^{-1})\nabla^2\Phi_k\right]^{-1}\nabla\Phi_k
\end{aligned}$$

tem-se

$$\mathbf{C}_k = \nabla^2\Phi_k^{-1}\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\mathbf{B}_k \tag{5.17}$$

em que

$$\mathbf{B}_k = \left[I - o_k\epsilon_k\nabla\Phi_k^T\right]. \tag{5.18}$$

5.5 Análise da convergência local

Considerem-se no conjunto aberto $D \subset \mathbb{R}^n$, a função $\Phi : D \subset \mathbb{R}^n \to \mathbb{R}$, definida por (5.3), o pseudo-gradiente $D\Phi(x)$ definido por (5.11) e a pseudo-Hessiana por (5.12). Enuncia-se de seguida um teorema que estabelece as condições para que a sucessão $\{x_k\}$ de aproximações à solução x^*, de $(P_{s/rest})$, gerada pela equação iterativa (5.13), seja bem definida e convirja localmente para x^*. Este teorema estabelece condições semelhantes às enunciadas no trabalho de Kelley [66] para outros métodos do tipo Newton.

A norma $\|.\|$ representa uma qualquer norma para vectores ou matrizes.

Teorema 5.1 *Sejam $\Phi : D \subset \mathbb{R}^n \to \mathbb{R}$ a função dada por (5.3) e $x_0 \in \mathcal{B}_\delta(x^*) \equiv \{x : \|x - x^*\| < \delta\} \subset L \subset D$ com $\delta > 0$ e L um conjunto compacto. Considerem-se verificadas em $\mathcal{B}_\delta(x^*)$ as Hipóteses 5.1 e 5.2, bem como*

i) *O operador $\epsilon(x) \in \mathbb{R}^n$ é contínuo à Lipschitz em x^* com constante γ_ϵ $(0 < \gamma_\epsilon < \infty)$, ou seja, como $\epsilon(x^*) = 0$,*

$$\|\epsilon(x_k)\| \leq \gamma_\epsilon \|x_k - x^*\|,$$

com $x_k \in \mathcal{B}_\delta(x^*)$;

ii) *O operador $\Delta(x) \in \mathbb{R}^{n\times n}$ é limitado*

$$\|\Delta(x)\| \leq \frac{1}{4\left\|\nabla^2\Phi(x^*)^{-1}\right\|};$$

iii) *Em x^*, $\nabla\Phi(x^*) = 0$;*

iv) *O operador $\nabla^2\Phi(x) \in \mathbb{R}^{n\times n}$ é contínuo à Lipschitz com constante K $(0 < K < \infty)$.*

Então a sucessão $\{x_k\}$ gerada pela equação iterativa do sistema pseudo-Newton

$$D^2\Phi(x_k)d_{PN_k} = -D\Phi(x_k), \quad k = 0, 1, \ldots,$$

com $D\Phi(x_k)$ definido por (5.11) e $D^2\Phi(x)$ por (5.12), é bem definida, converge para x^* pelo menos q-linearmente e obedece a

$$\left\|e_{k+1}^{PN}\right\| \leq \left\|e_{k+1}^{N}\right\| + 4\beta\left(\nabla^2\Phi(x^*) + \gamma_\epsilon\right)\|e_k\|$$

e

$$\left\|e_{k+1}^{PN}\right\| \leq \left[\beta K\delta + 4\beta\left(\left\|\nabla^2\Phi(x^*)\right\| + \gamma_\epsilon\right)\right]\|e_k\| < \|e_k\|,$$

em que $e_k = x_k - x^*$.

Prova. Na demonstração vai usar-se a notação A_k para designar qualquer operador $A(x)$, calculado no iterando x_k. Considerem-se a iteração Newton

$$x_{k+1}^{N} = x_k - \nabla^2\Phi_k^{-1}\nabla\Phi_k$$

e a pseudo-Newton

$$x_{k+1}^{PN} = x_k - \left[\nabla^2\Phi_k - \Delta_k\right]^{-1}\left[\nabla\Phi_k - \epsilon_k\right].$$

Subtraindo ao próximo iterando obtido pelo método pseudo-Newton, o iterando seguinte do método de Newton vem

$$x_{k+1}^{PN} - x_{k+1}^{N} = -\left[\nabla^2\Phi_k - \Delta_k\right]^{-1}\left[\nabla\Phi_k - \epsilon_k\right] + \nabla^2\Phi_k{}^{-1}\nabla\Phi_k.$$

Subtraindo x^* a ambos os membros,

$$x_{k+1}^{PN} - x^* = x_{k+1}^{N} - x^* + \left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\nabla\Phi_k + \left[\nabla^2\Phi_k - \Delta_k\right]^{-1}\epsilon_k,$$

usando a definição de erro da iteração $k+1$, $e_{k+1} = x_{k+1} - x^*$ e tomando normas tem-se

$$\left\|e_{k+1}^{PN}\right\| \leq \left\|e_{k+1}^{N}\right\| + \left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\|\|\nabla\Phi_k\| + \left\|\left[\nabla^2\Phi_k - \Delta_k\right]^{-1}\right\|\|\epsilon_k\|.$$

Utilizando a razão de convergência do método de Newton ([66, pp 71]), as condições ii) e iii) do Lema 2.1, com as correspondentes alterações, (5.16) e a condição i), obtém-se

$$\begin{aligned}
\left\|e_{k+1}^{PN}\right\| &\leq \left\|e_{k+1}^{N}\right\| + 2\left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\|\left\|\nabla^2\Phi^*\right\|\|e_k\| + \\
&\quad + \left\|\left[\nabla^2\Phi_k^{-1}(I - \nabla^2\Phi_k{}^{-1}\Delta_k)\right]^{-1}\right\|\gamma_\epsilon\|e_k\| \\
&\leq \beta K\|e_k\|^2 + 2\left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\|\left\|\nabla^2\Phi^*\right\|\|e_k\| + \\
&\quad + \left\|[I - \nabla^2\Phi_k{}^{-1}\Delta_k]^{-1}\right\|2\beta\gamma_\epsilon\|e_k\|.
\end{aligned}$$

Analisando o termo $\left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\|$ separadamente tem-se

$$\begin{aligned}\left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\| &\leq \frac{\left\|\nabla^2\Phi_k{}^{-1}\right\| \left\|I - \left(\nabla^2\Phi_k - \Delta_k\right)\nabla^2\Phi_k{}^{-1}\right\|}{1 - \left\|I - \left(\nabla^2\Phi_k - \Delta_k\right)\nabla^2\Phi_k{}^{-1}\right\|} \\ &\leq \frac{2\beta\left\|\Delta_k\nabla^2\Phi_k{}^{-1}\right\|}{1 - \left\|\Delta_k\nabla^2\Phi_k{}^{-1}\right\|}\end{aligned}$$

por aplicação do Teorema B.5 no Apêndice B (Teorema de Banach), condição ii) do Lema 2.1 e Hipótese 5.1. Pela condição ii) e usando novamente a condição ii) do Lema 2.1 vem

$$\begin{aligned}\|\Delta_k\| &\leq \frac{1}{4\left\|\nabla^2\Phi^{*-1}\right\|} \\ &\leq \frac{1}{2\left\|\nabla^2\Phi_k{}^{-1}\right\|} \Leftrightarrow \\ &\Leftrightarrow \|\Delta_k\|\left\|\nabla^2\Phi_k{}^{-1}\right\| \leq \frac{1}{2} \Leftrightarrow \\ &\Leftrightarrow \left\|\Delta_k\nabla^2\Phi_k{}^{-1}\right\| \leq \frac{1}{2}.\end{aligned}$$

Usando este resultado em $\dfrac{2\beta\left\|\Delta_k\nabla^2\Phi_k{}^{-1}\right\|}{1 - \left\|\Delta_k\nabla^2\Phi_k{}^{-1}\right\|}$ obtém-se

$$\left\|\left[\nabla^2\Phi_k{}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\| \leq 2\beta.$$

Substituindo fica

$$\left\|e_{k+1}^{PN}\right\| \leq \beta K\|e_k\|^2 + 4\beta\left\|\nabla^2\Phi^*\right\|\|e_k\| + \left\|[I - \nabla^2\Phi_k^{-1}\Delta_k]^{-1}\right\| 2\beta\gamma_\epsilon\|e_k\|.$$

Como $\left\|\nabla^2\Phi_k{}^{-1}\Delta_k\right\| \leq \dfrac{1}{2}$ pode aplicar-se o Lema B.7 no Apêndice B ao termo $\left\|[I - \nabla^2\Phi_k{}^{-1}\Delta_k]^{-1}\right\|$ ficando

$$\left\|[I - \nabla^2\Phi_k{}^{-1}\Delta_k]^{-1}\right\| \leq \frac{1}{1 - \left\|\nabla^2\Phi_k{}^{-1}\Delta_k\right\|} \leq 2$$

e substituindo mais uma vez vem

$$\begin{aligned}\left\|e_{k+1}^{PN}\right\| &\leq \beta K\|e_k\|^2 + 4\beta\left\|\nabla^2\Phi^*\right\|\|e_k\| + 4\beta\gamma_\epsilon\|e_k\| \\ &\leq \left[\beta K\delta + 4\beta\left\|\nabla^2\Phi^*\right\| + 4\beta\gamma_\epsilon\right]\|e_k\|.\end{aligned}$$

Se δ for reduzido de forma a que $\beta\left(K\delta + 4\left\|\nabla^2\Phi^*\right\| + 4\gamma_\epsilon\right) < 1$ então $x_k \to x^*$ q-linearmente. ■

Corolário 5.1 *Considerem-se as hipóteses do teorema anterior. Se além disso, as restrições forem lineares e a sucessão* $\{x_k\}$ *for tal que* $c_i(x_k) = 0$, $i = 1, \ldots, m$ *então o método converge q-quadraticamente.*

Prova. Usando alguns resultados da prova do teorema anterior tem-se

$$\begin{aligned}
\left\|e_{k+1}^{PN}\right\| &\leq \left\|e_{k+1}^{N}\right\| + \left\|\left[{\nabla^2\Phi_k}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\| \|\nabla\Phi_k\| + \left\|\left[\nabla^2\Phi_k - \Delta_k\right]^{-1}\right\| \|\epsilon_k\| \\
&\leq \beta K \|e_k\|^2 + 2\left\|\nabla^2\Phi^*\right\| \|e_k\| \left\|\left[{\nabla^2\Phi_k}^{-1} - \left(\nabla^2\Phi_k - \Delta_k\right)^{-1}\right]\right\| + \\
&\quad + \left\|\nabla^2\Phi_k^{-1}(I - \nabla^2\Phi_k^{-1}\Delta_k)^{-1}\right\| \|\epsilon_k\| \\
&\leq \beta K \|e_k\|^2 + 2\left\|\nabla^2\Phi^*\right\| \|e_k\| \frac{2\beta\left\|\Delta_k{\nabla^2\Phi_k}^{-1}\right\|}{1 - \left\|\Delta_k{\nabla^2\Phi_k}^{-1}\right\|} + 2\beta\left\|(I - \nabla^2\Phi_k^{-1}\Delta_k)^{-1}\right\| \|\epsilon_k\| \\
&\leq \beta K \|e_k\|^2 + 4\beta\left\|\nabla^2\Phi^*\right\| \|e_k\| \|\Delta_k\| \frac{\left\|{\nabla^2\Phi_k}^{-1}\right\|}{1 - \left\|\Delta_k{\nabla^2\Phi_k}^{-1}\right\|} + 4\beta\|\epsilon_k\| \\
&\leq \beta K \|e_k\|^2 + 16\beta^2\left\|\nabla^2\Phi^*\right\| \|e_k\| \|\Delta_k\| + 4\beta\|\epsilon_k\| \\
&\leq \widehat{K}(\|e_k\|^2 + \|\Delta_k\| \|e_k\| + \|\epsilon_k\|),
\end{aligned}$$

com $\widehat{K} = \beta K + 16\beta^2\left\|\nabla^2\Phi^*\right\| + 4\beta$. Se x_k estiver na região admissível então $\epsilon_k = 0$ e se, para além disso, $c_i(x)$, $i = 1, \ldots, m$, forem funções lineares tem-se $\Delta_k = 0$. Assim obtém-se

$$\left\|e_{k+1}^{PN}\right\| \leq \widehat{K}(\|e_k\|^2)$$

e $\{x_k\}$ converge q-quadraticamente para x^*. ■

5.6 Direcção pseudo-Newton modificada

O Capítulo 2 foi dedicado ao estudo da direcção Newton modificada, no âmbito da optimização não linear sem restrições. Da mesma forma, para ultrapassar os eventuais problemas da direcção pseudo-Newton, vai agora apresentar-se uma nova direcção, a direcção pseudo-Newton modificada, denotada por d_{PNM}.

Dependendo da matriz $\mathbf{C}$ (veja-se em (5.17)), no iterando x_k, assim, a direcção pseudo-Newton, d_{PN_k}, pode não ser uma direcção descendente em relação a Φ, pode ser ortogonal ao gradiente ou pode até nem ser definida. Para ultrapassar estes inconvenientes é possível definir-se uma nova direcção modificada, d_{PNM}, combinando a direcção pseudo-Newton

com uma direcção que é denotada por pseudo-descida máxima, dada por $-D\Phi_k$. Define-se, assim,

$$d_{PNM_k} = -(1 - w(x_k))D^2\Phi_k^{-1}D\Phi_k - w(x_k)D\Phi_k \tag{5.19}$$

com $w(x_k)$ uma função real de variável real definida por

$$w(x_k) = \frac{D\Phi_k^T d_{PN_k} + \varepsilon \|D\Phi_k\|_2^2}{\|D\Phi_k\|_2^2 + D\Phi_k^T d_{PN_k}} \tag{5.20}$$

para $\varepsilon \in (0, 1]$, garantindo-se $D\Phi_k^T\, d_{PNM_k} < 0$. Como $D\Phi_k = \mathbf{B}_k\nabla\Phi_k$ (veja-se em (5.15) e (5.18)), tem-se

$$w(x_k) = \frac{\nabla\Phi_k^T\mathbf{B}_k^T d_{PN_k} + \varepsilon \|\mathbf{B}_k\nabla\Phi_k\|_2^2}{\|\mathbf{B}_k\nabla\Phi_k\|_2^2 + \nabla\Phi_k^T\mathbf{B}_k^T d_{PN_k}}$$

em função de $\nabla\Phi_k$.

Para direcções pseudo-Newton suficientemente pseudo-descendentes, isto é, que verifiquem

$$D\Phi_k^T\, d_{PN_k} \leq -\overline{\eta}_1 \|D\Phi_k\| \|d_{PN_k}\| \tag{5.21}$$

com $\overline{\eta}_1 > 0$, tem-se $d_{PNM_k} = d_{PN_k}$ $(w(x_k) = 0)$. Para matrizes $D^2\Phi_k$ singulares, define-se $d_{PNM_k} = -D\Phi_k$, que equivale a fazer $w(x_k) = 1$ em (5.19). Para os outros casos, calcula-se a função $w(x_k)$ (de (5.20)) e a direcção modificada é obtida a partir de (5.19). De acordo com a definição, $w(x_k) \in [0, 1]$.

Seguindo um raciocínio idêntico ao da secção 5.4, a direcção pseudo-Newton modificada também pode ser reescrita em função de $\nabla\Phi_k$:

$$\begin{aligned} d_{PNM_k} &= -(1 - w_k)\mathbf{C}_k\nabla\Phi_k - w_k D\Phi_k \\ &= -\{(1 - w_k)\mathbf{C}_k\nabla\Phi_k + w_k(\nabla\Phi_k - \epsilon_k)\} \\ &= -\left\{(1 - w_k)\mathbf{C}_k\nabla\Phi_k + w_k\nabla\Phi_k - w_k\epsilon_k(o_k\nabla\Phi_k^T\nabla\Phi_k)\right\} \end{aligned}$$

tomando w_k em vez de $w(x_k)$ e considerando $o_k = \dfrac{1}{\|\nabla\Phi_k\|_2^2}$. Assim

$$\begin{aligned} d_{PNM_k} &= -\left\{(1 - w_k)\mathbf{C}_k + w_k\left[I - o_k\epsilon_k\nabla\Phi_k^T\right]\right\}\nabla\Phi_k \\ &= -\{(1 - w_k)\mathbf{C}_k + w_k\mathbf{B}_k\}\nabla\Phi_k \qquad (5.22) \\ &= -\left\{(1 - w_k)\nabla^2\Phi_k^{-1}\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\mathbf{B}_k + w_k\mathbf{B}_k\right\}\nabla\Phi_k. \qquad (5.23) \end{aligned}$$

O algoritmo que corresponde ao procedimento que calcula a direcção pseudo-Newton modificada baseada em (5.19) é o seguinte:

Algoritmo 5.2 (Método Pseudo-Newton modificado)

Dados $\overline{\eta}_1$, $\overline{\mu}_1$, δ_1 e x_0;
Enquanto *(Critério de paragem = falso)* ***Fazer***
  {Calcular d_{PNM_k} :
    Se *(*$D^2\Phi_k$ *é singular)* ***Então*** $w_k = 1$;
    Senão *{calcular* d_{PN_k}*, solução de* $D^2\Phi_k d_{PN_k} = -D\Phi_k$*};*
      Se *(*$D\Phi_k^T d_{PN_k} \leq -\overline{\eta}_1 \|D\Phi_k\| \, \|d_{PN_k}\|$*)* ***Então*** $w_k = 0$;
      Senão *{calcular* w_k *usando (5.20)}*
      FimSe
    FimSe
    Calcular d_{PNM_k} *usando (5.19)}*
  $\{\alpha = 1; \quad x = x_k + \alpha_k d_{PNM_k}\}$
  Enquanto $\Phi(x_k) - \Phi(x) < -\overline{\mu}_1 \alpha D\Phi_k^T d_{PNM_k}\}$ ***Fazer***
    Se $(\|\alpha d_{PNM_k}\|_2 > \delta_1)$ ***Então***
      $\left\{\alpha = \dfrac{\alpha}{2}; \quad x = x_k + \alpha d_{PNM_k}\right\}$
    Senão
      $\{\alpha = 1; \quad x = x_k + \alpha d_{PNM_k}\}$
      $\{SAIR\}$
    FimSe
  FimEnquanto
  $\{x_{k+1} = x\}$
FimEnquanto

Comecemos por analisar a matriz $\mathbf{B}_k$, definida em (5.18), que surge também na definição da direcção pseudo-Newton modificada, d_{PNM_k}, em (5.23).

A matriz $\epsilon_k \nabla\Phi_k^T$ tem característica um, logo tem $(n-1)$ valores próprios nulos e um valor próprio igual a $tr(\epsilon_k \nabla\Phi_k^T)$[1]. Se o vector ϵ_k for $(\epsilon_{(1)}\epsilon_{(2)} \ldots \epsilon_{(n)})^T$ e $\nabla\Phi_k = (\nabla\Phi_{(1)}\nabla\Phi_{(2)} \ldots \nabla\Phi_{(n)})^T$ então a matriz $\epsilon_k \nabla\Phi_k^T$ tem a seguinte forma

$$\begin{bmatrix} \epsilon_{(1)}\nabla\Phi_{(1)} & \epsilon_{(1)}\nabla\Phi_{(2)} & \cdots & \epsilon_{(1)}\nabla\Phi_{(n)} \\ \epsilon_{(2)}\nabla\Phi_{(1)} & \epsilon_{(2)}\nabla\Phi_{(2)} & & \epsilon_{(2)}\nabla\Phi_{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{(n)}\nabla\Phi_{(1)} & \epsilon_{(n)}\nabla\Phi_{(2)} & \cdots & \epsilon_{(n)}\nabla\Phi_{(n)} \end{bmatrix}$$

[1]Para qualquer matriz A quadrada, $tr(A)$ denota traço da matriz.

e $tr(\epsilon_k \nabla \Phi_k^T) = \epsilon_k^T \nabla \Phi_k$. Logo, o valor próprio não nulo é

$$vp\left(\epsilon_k \nabla \Phi_k^T\right) = \epsilon_k^T \nabla \Phi_k \tag{5.24}$$

em que *vp* denota valor próprio.

Definição 5.4 *Se* $\Phi : D \subset \mathbb{R}^n \to \mathbb{R}$, *então qualquer conjunto não vazio da forma*

$$L^0 = \{x \in D : \Phi(x) \leq \Phi(x_0)\}$$

é o conjunto de nível de Φ.

Hipótese 5.3 *Dada uma aproximação inicial* $x_0 \in L^0$, *suponha-se que o conjunto de nível* L^0 ($L^0 \subset D$) *é compacto.*

De seguida, vai ser enunciada uma hipótese para garantir a não singularidade da matriz $\mathbf{B}_k$.

Hipótese 5.4 *A matriz* $o_k \epsilon_k \nabla \Phi_k^T$ *é convergente, isto é,*

$$\left|vp(o_k \epsilon_k \nabla \Phi_k^T)\right| < 1,$$

para $x_k \in L^0$ *em que vp denota o valor próprio não nulo da matriz.*

De acordo com (5.24), esta hipótese implica

$$\begin{aligned} \left|o_k \epsilon_k^T \nabla \Phi_k\right| &< 1 \\ \left|\epsilon_k^T \nabla \Phi_k\right| &< \|\nabla \Phi_k\|_2^2. \end{aligned} \tag{5.25}$$

Da Hipótese 5.4 resulta que $I - o_k \epsilon_k \nabla \Phi_k^T$ é não singular [72, pp119].

A matriz $\mathbf{B}$, não simétrica, é definida positiva ($\forall x \in \mathbb{R}^n, x \neq 0,\ x^T \mathbf{B} x > 0$) se e só se a matriz simétrica $\mathbf{B} + \mathbf{B}^T$ for definida positiva ([88, pp35]). Seja $\lambda_1(A)$ o menor valor próprio de A. Como

$$\begin{aligned} x^T(\mathbf{B}+\mathbf{B^T})x &= x^T(2I - o(\epsilon \nabla \Phi^T + \nabla \Phi \epsilon^T))x \\ &\geq \lambda_1(2I - o(\epsilon \nabla \Phi^T + \nabla \Phi \epsilon^T)x^T x \\ &= (2 - \lambda_1(o(\epsilon \nabla \Phi^T + \nabla \Phi \epsilon^T)))x^T x \\ &\geq (2 - 2o \|\epsilon\|_2 \|\nabla \Phi\|_2)x^T x \\ &= 2(1 - \frac{\|\epsilon\|_2}{\|\nabla \Phi\|_2})x^T x \\ &> 0 \end{aligned}$$

se $\|\epsilon\|_2 < \|\nabla\Phi\|_2$. Por absurdo, suponha-se que $\|\epsilon\|_2 \geq \|\nabla\Phi\|_2$. Da desigualdade de Cauchy Schwarz $\left|\epsilon^T\nabla\Phi\right| \leq \|\epsilon\|_2 \|\nabla\Phi\|_2$, ou seja,

$$\frac{\left|\epsilon^T\nabla\Phi\right|}{\|\nabla\Phi\|_2^2} \leq \frac{\|\epsilon\|_2}{\|\nabla\Phi\|_2} = a$$

com $a \geq 1$, o que contraria (5.25) que é uma consequência imediata da Hipótese 5.4. Portanto, tem-se sempre $\|\epsilon\|_2 < \|\nabla\Phi\|_2$.

Considere-se ainda a seguinte proposição.

Proposição 5.3 *Seja h um vector de $\mathbb{R}^n$. De acordo com as Hipóteses 5.3 e 5.4 e para a matriz $\mathbf{B}_k$ definida por (5.18) tem-se*

$$\overline{\epsilon}h^Th \leq h^T\mathbf{B}_kh < 2h^Th \tag{5.26}$$

para $\overline{\epsilon}$ pequeno e positivo.

5.7 Estudo da convergência global

Nesta secção vai ser feito o estudo da convergência global do método pseudo-Newton modificado. Vão ser consideradas algumas hipóteses e apresentados alguns lemas e teoremas necessários, cuja demonstração, quando não desenvolvida, é remetida para a bibliografia respectiva.

Nesta secção $\|.\|$ denota uma qualquer norma arbitrária em $\mathbb{R}^n$ e a correspondente norma do operador induzida.

Considerem-se as seguintes hipóteses.

Considerando a Hipótese 5.1 e devido à continuidade de $\nabla^2\Phi^{-1}(x)$ no compacto L^0, tem-se que existe um escalar positivo k_2 tal que

$$\left\|\nabla^2\Phi_k^{-1}\right\| \leq k_2\beta, \tag{5.27}$$

para todo o $x_k \in L^0$.

Considere-se agora a função $\epsilon(x)$ definida na Secção 5.4.

Hipótese 5.5 *O operador $\epsilon(x) : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ é contínuo à Lipschitz em x^*, de constante γ_ϵ positiva ($0 < \gamma_\epsilon < \infty$), isto é,*

$$\|\epsilon(x_k) - \epsilon(x^*)\| \leq \gamma_\epsilon \|x_k - x^*\|,$$

para x_k, $x^* \in L^0$.

Como

$$\begin{aligned} \epsilon(x^*) &= [D\lambda(x^*) - \nabla\lambda(x^*)]^T c(x^*) \\ &= 0 \end{aligned}$$

(veja-se em (5.15) na secção 5.4), tem-se que

$$\|\epsilon(x_k)\| \leq \gamma_\epsilon \|x_k - x^*\|. \tag{5.28}$$

Tome-se a matriz $\Delta(x)$ definida na secção 5.4.

Hipótese 5.6 *O operador* $\Delta(x) : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$ *é contínuo à Lipschitz em* x^*, *de constante* γ_Δ *positiva* $(0 < \gamma_\Delta < \infty)$, *isto é,*

$$\|\Delta(x_k) - \Delta(x^*)\| \leq \gamma_\Delta \|x_k - x^*\|,$$

para x_k, $x^* \in L^0$.

Hipótese 5.7 *A matriz* $\Delta(x)$, *no ponto* x^*, *é limitada em norma*

$$\|\Delta(x^*)\| \leq \alpha_\Delta, \quad (0 < \alpha_\Delta < \infty).$$

Hipótese 5.8 *O operador* $\nabla\Phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ *é contínuo à Lipschitz em* L^0, *de constante* $\gamma_{\nabla\Phi}$ $(1 < \gamma_{\nabla\Phi} < \infty)$, *isto é,*

$$\|\nabla\Phi(x_{k+1}) - \nabla\Phi(x_k)\| \leq \gamma_{\nabla\Phi} \|x_{k+1} - x_k\|,$$

para $x_{k+1}, x_k \in L^0$.

Da Hipótese 5.8 e como consequência da continuidade e da não singularidade de $\nabla^2\Phi$, no ponto x^* (Hipótese 5.1), tem-se (veja-se [88])

$$\frac{1}{\gamma_{\nabla\Phi}} \|x_k - x^*\| \leq \|\nabla\Phi(x_k)\| \leq \gamma_{\nabla\Phi} \|x_k - x^*\| \tag{5.29}$$

pois $\nabla\Phi(x^*) = 0$.

Hipótese 5.9 *O operador* $\nabla^2\Phi(x) : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$ *é contínuo à Lipschitz em* L^0, *de constante* K *positiva* $(0 < K < \infty)$, *isto é,*

$$\left\|\nabla^2\Phi(x_{k+1}) - \nabla^2\Phi(x_k)\right\| \leq K \|x_{k+1} - x_k\|, \tag{5.30}$$

para $x_{k+1}, x_k \in L^0$.

As matrizes $\mathbf{B}_k$ e $\mathbf{C}_k$ em d_{PN_k} e d_{PNM_k} vão ser estudadas de seguida.

Por uma questão de simplificação da notação, a partir de agora d_{PNM_k} é designada por d_k.

Lema 5.2 *Considerem-se as Hipóteses 5.1, 5.5 e 5.8. Seja* $\widetilde{\delta} > 0$. *Então a matriz* $\mathbf{B}_k$ *definida por (5.18) satisfaz*

$$1 - \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi} \leq \|\mathbf{B}_k\| \leq 1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}.$$

Prova. De acordo com a definição (5.18)

$$1 - o_k \left\|\epsilon_k \nabla\Phi_k^T\right\| \leq \|\mathbf{B}_k\| \leq 1 + o_k \left\|\epsilon_k \nabla\Phi_k^T\right\|.$$

Como

$$\left\|\epsilon_k \nabla\Phi_k^T\right\| \leq \|\epsilon_k\| \, \|\nabla\Phi_k\|$$

tem-se

$$\left\|\epsilon_k \nabla\Phi_k^T\right\| \leq \gamma_\epsilon \|x_k - x^*\| \, \|\nabla\Phi_k\|$$

usando (5.28). Assim

$$1 - o_k\gamma_\epsilon \|x_k - x^*\| \, \|\nabla\Phi_k\| \leq \|\mathbf{B}_k\| \leq 1 + o_k\gamma_\epsilon \|x_k - x^*\| \, \|\nabla\Phi_k\|$$

e como $\|\nabla\Phi_k\| \leq \widetilde{\delta} \|\nabla\Phi_k\|_2$ para $0 < \widetilde{\delta} < \infty$ (pelo Teorema B.1 no Apêndice B), tem-se

$$1 - \delta_k \leq \|\mathbf{B}_k\| \leq 1 + \delta_k \tag{5.31}$$

para $\delta_k = \widetilde{\delta}\gamma_\epsilon \|x_k - x^*\| \left(\|\nabla\Phi_k\|_2\right)^{-1}$. Por (5.29)

$$\delta_k \leq \widetilde{\delta}\gamma_\epsilon \|x_k - x^*\| \left(\frac{1}{\gamma_{\nabla\Phi}} \|x_k - x^*\|\right)^{-1}$$

donde

$$1 - \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi} \leq \|\mathbf{B}_k\| \leq 1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}. \tag{5.32}$$

■

Lema 5.3 *Considerem-se as Hipóteses 5.1, 5.2, 5.4, 5.5, 5.6, 5.7 e 5.8. Seja k_3 uma constante maior ou igual a um,*

$$\xi_1 = k_2\beta \qquad e \qquad \xi_2 = 1 + k_2\beta(\alpha_\Delta + \gamma_\Delta\sigma), \tag{5.33}$$

quantidades positivas e $\sigma = máx\,\|x_k - x^*\|$, para $x_k \in L^0$. Então a matriz $\mathbf{C}_k$ definida por (5.17) é limitada superiormente em norma

$$\|\mathbf{C}_k\| \leq \xi_1 k_3 \xi_2^{-1}(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}).$$

Prova. Se a matriz $\mathbf{C}_k$ for posta na forma $\mathbf{C}_k = D^2\Phi_k^{-1}\mathbf{B}_k$ (veja-se (5.17)), basta encontrar um limite para $\left\|D^2\Phi_k^{-1}\right\|$. Como pela Hipótese 5.2

$$\left\|D^2\Phi_k^{-1}\right\| = \left\|\nabla^2\Phi_k^{-1}\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\right\| \leq \left\|\nabla^2\Phi_k^{-1}\right\| \left\|\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\right\|$$

e

$$\left\|I - \Delta_k\nabla^2\Phi_k^{-1}\right\| \leq 1 + \|\Delta_k\| \left\|\nabla^2\Phi_k^{-1}\right\|,$$

tem-se pelas Hipóteses 5.6 e 5.7 e por (5.27)

$$\left\|I - \Delta_k\nabla^2\Phi_k^{-1}\right\| \leq 1 + (\alpha_\Delta + \gamma_\Delta\sigma)k_2\beta$$

com $\sigma = máx\,\|x_k - x^*\|$. Como para qualquer matriz A, $\|A^{-1}\| \geq (\|A\|)^{-1}$, vem

$$\left\|\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\right\| \geq (1 + (\alpha_\Delta + \gamma_\Delta\sigma)k_2\beta)^{-1}.$$

Nestas condições existe uma constante $k_3 \geq 1$, tal que

$$\left\|\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\right\| \leq k_3\xi_2^{-1}$$

sendo $\xi_2 = 1+(\alpha_\Delta+\gamma_\Delta\sigma)k_2\beta$ uma quantidade positiva. O resultado final agora é imediato:

$$\begin{aligned} \|\mathbf{C}_k\| &\leq \left\|\nabla^2\Phi_k^{-1}\right\| \left\|\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\right\| \|\mathbf{B}_k\| \\ &\leq \xi_1 k_3 \xi_2^{-1}\left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right). \end{aligned} \tag{5.34}$$

■

Para a análise que se segue considere-se a direcção pseudo-Newton modificada, d_k, reescrita na forma

$$d_k = -\left\{(1 - w_k)\mathbf{C}_k + w_k\mathbf{B}_k\right\}\nabla\Phi_k \tag{5.35}$$

com $\mathbf{C}_k = \nabla^2\Phi_k^{-1}\left(I - \Delta_k\nabla^2\Phi_k^{-1}\right)^{-1}\mathbf{B}_k$ e $\mathbf{B}_k = \left[I - o_k\epsilon_k\nabla\Phi_k^T\right]$.

Lema 5.4 *Sejam* $\widehat{\gamma}_1 = max(0, \gamma_1)$ *e* η_1 *positivo. Então para* $\gamma_1 \leq \dfrac{2}{(\eta_1 + 2)}$ *tem-se*

$$0 \leq \frac{1}{\eta_1 + 2} - \frac{1}{2}\widehat{\gamma}_1 < 1. \tag{5.36}$$

Prova. Para $\gamma_1 < 0$, tem-se $\widehat{\gamma}_1 = 0$ e (5.36) reduz-se a

$$0 \leq \frac{1}{\eta_1 + 2} < 1.$$

Para $\gamma_1 > 0$, tem-se $\widehat{\gamma}_1 = \gamma_1$, donde

$$\frac{1}{\eta_1 + 2} - \frac{1}{2}\gamma_1 < 1$$

implica

$$2 - \gamma_1(\eta_1 + 2) - 2(\eta_1 + 2) \quad < 0.$$

Esta desigualdade exige que

$$\gamma_1 > \frac{-2(\eta_1 + 1)}{(\eta_1 + 2)}$$

que é sempre verdadeiro. Da mesma forma

$$0 \leq \frac{1}{(\eta_1 + 2)} - \frac{1}{2}\gamma_1$$

implica que $2 - (\eta_1 + 2)\gamma_1 \geq 0$ que é verdadeiro sempre que $\gamma_1 \leq \dfrac{2}{\eta_1 + 2}$. ■

Lema 5.5 *Considerem-se as Hipóteses 5.1, 5.2, 5.4, 5.5, 5.6, 5.7 e 5.8. Para* $w_k \in [0, 1)$ *e* d_k *definida por (5.35)*

$$\|d_k\| < \left(1 + \widetilde{\delta}\gamma_\epsilon \gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)\|\nabla\Phi_k\|$$

em que $\widetilde{\delta}$ *e* $k_3 \geq 1$, ξ_1 *e* ξ_2 *foram definidos respectivamente nos Lemas 5.2 e 5.3.*

Prova. Usando a definição (5.35) e as relações (5.32) e (5.34) tem-se

$$\begin{aligned}
\|d_k\| &\leq \{(1-w_k)\|\mathbf{C}_k\| + w_k\|\mathbf{B}_k\|\}\|\nabla\Phi_k\| \\
&\leq \left\{(1-w_k)\xi_1 k_3 \xi_2^{-1}\left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right) + w_k\left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\right\}\|\nabla\Phi_k\| \\
&< \left\{\xi_1 k_3\xi_2^{-1}\left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right) + \left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\right\}\|\nabla\Phi_k\| \\
&< \left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\left(\xi_1 k_3\xi_2^{-1}+1\right)\|\nabla\Phi_k\|. \qquad (5.37)
\end{aligned}$$

■

O teorema seguinte estabelece condições para que a direcção pseudo-Newton modificada satisfaça algumas propriedades. Como se verá, sob certas condições, a direcção d_k é suficientemente descendente em relação a Φ, no ponto x_k, é considerada do tipo gradiente, é limitada em norma e garante uma redução significativa no valor da função Φ, ao longo do processo iterativo.

Teorema 5.2 *Considerem-se as Hipóteses 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8. Seja* $\Phi : L^0 \subset D \subset \mathbb{R}^n \to \mathbb{R}$ *uma função continuamente diferenciável e suponha que* Φ *tem um número finito de pontos estacionários em* L^0. *Suponhamos ainda que* $(1-w_k)\mathbf{C}_k : L^0 \longrightarrow \mathbb{R}^{n\times n}$, *com* $\mathbf{C}_k$ *definida por (5.17) e* $w_k \in [0,1)$, *é contínua para todo o* $x_k \in L^0$ *e que*

$$\eta_0\overline{\epsilon}h^Th \leq h^T(1-w_k)\mathbf{C}_k h \leq \eta_1 h^T h, \qquad \forall x_k \in L^0, h \in \mathbb{R}^n \qquad (5.38)$$

$$h^T\nabla^2\Phi_k h \leq \gamma_1 h^T h \qquad (5.39)$$

e $\widehat{\gamma}_1 = max(0,\gamma_1)$, *com* γ_1 *não necessariamente positivo,* $\eta_1 = \xi_1 k_3\xi_2^{-1}\left(1+\widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)$ *com* ξ_1 *e* ξ_2 *definidos por (5.33) e* $k_3 \geq 1$. *Seja* $\{x_k\}$ *a sucessão definida por* $x_{k+1} = x_k + \alpha_k d_k$ *com* d_k *a direcção definida por (5.35). Suponha que*

$$\frac{1}{2}\widehat{\gamma}_1 - \eta_0 < l_0 \leq w_k \leq \frac{1}{\eta_1+2} - \frac{1}{2}\widehat{\gamma}_1 < 1. \qquad (5.40)$$

onde

$$0 \leq l_0 < 1-\eta_0, \qquad (5.41)$$

$\gamma_1 \leq \dfrac{2}{(\eta_1+2)}$ *e* $\bar{\epsilon}$ *pequeno e positivo (veja-se (5.26)). Então verificam-se as seguintes propriedades:*

$$i) \quad -\frac{\nabla\Phi_k^T d_k}{\|\nabla\Phi_k\|\,\|d_k\|} > \varphi > 0;$$

$$ii) \quad \|d_k\| \geq N\,\|\nabla\Phi_k\| \;\; com\; N > 0;$$

$$iii) \quad \|d_k\| < M \;\; com\; M > 0;$$

$$iv) \quad \Phi(x_k + \alpha_k d_k) < \Phi(x_k) + \mu\alpha_k \nabla\Phi_k^T d_k, \;\; com\; 0 < \mu < 1 \;\; e\; \alpha_k \in (0,1];$$

$$v) \quad \{x_k\} \subset L^0.$$

Prova. Esta demonstração vai ser feita separadamente para cada uma das propriedades enunciadas. Para provar i) considere-se a definição de d_k em (5.35) e

$$-\nabla\Phi_k^T d_k = \nabla\Phi_k^T \left\{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\right\} \nabla\Phi_k.$$

Como $w_k \in [l_0, 1)$, de (5.26) e (5.38) para qualquer $h \in \mathbb{R}^n$ tem-se

$$(\eta_0 + l_0)\bar{\epsilon} h^T h \leq h^T((1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k)\, h < (\eta_1+2)\, h^T h, \; \forall h \in \mathbb{R}^n. \tag{5.42}$$

Em especial, para $h = \nabla\Phi_k$

$$-\nabla\Phi_k^T d_k \geq (\eta_0 + l_0)\bar{\epsilon}\,\|\nabla\Phi_k\|_2^2. \tag{5.43}$$

Substituindo e considerando (5.37) vem

$$\begin{aligned} -\frac{\nabla\Phi_k^T d_k}{\|\nabla\Phi_k\|\,\|d_k\|} &> \frac{(\eta_0+l_0)\bar{\epsilon}\,\|\nabla\Phi_k\|_2^2}{\|\nabla\Phi_k\|_2^2 \widetilde{\delta}^2 \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)} \\ &= \frac{(\eta_0+l_0)\bar{\epsilon}}{\widetilde{\delta}^2 \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)} = \varphi > 0 \end{aligned}$$

pois de (5.40) $0 \leq \frac{1}{2}\widehat{\gamma}_1 < (\eta_0 + l_0)$. Para provar ii) tome-se novamente

$$\begin{aligned} -\nabla\Phi_k^T d_k &= \nabla\Phi_k^T \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\} \nabla\Phi_k \\ &= \nabla\Phi_k^T \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\}^T \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\}^{T^{-1}} \\ &\quad \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\} \nabla\Phi_k \\ &= d_k^T \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\}^{T^{-1}} d_k \\ &= d_k^T \{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\}^{-1} d_k \\ &\leq ((\eta_0 + l_0)\overline{\epsilon})^{-1} \|d_k\|_2^2 \end{aligned} \tag{5.44}$$

por (5.42). Atendendo a (5.43) vem

$$\|d_k\|_2^2 \geq ((\eta_0 + l_0)\,\overline{\epsilon})^2 \|\nabla\Phi_k\|_2^2 \tag{5.45}$$

e $\|d_k\|_2 \geq N \|\nabla\Phi_k\|_2$ sendo $N = (\eta_0 + l_0)\overline{\epsilon} > 0$. Prova de iii): dada uma matriz invertível $\mathcal{E}$, tem-se para quaisquer $\mathcal{E}$, $x \in L^0$ e $h \in \mathbb{R}^n$

$$\begin{aligned} \|h\| &= \left\|\mathcal{E}\mathcal{E}^{-1}h\right\| \\ &\leq \|\mathcal{E}\| \left\|\mathcal{E}^{-1}h\right\|. \end{aligned}$$

Particularizando para $\mathcal{E} = (1-w)\mathbf{C} + w\mathbf{B}$ e usando (5.32) e (5.34) tem-se

$$\|h\| < \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right) \left\|\mathcal{E}^{-1}h\right\|$$

donde

$$\left\|\mathcal{E}^{-1}h\right\| > \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)^{-1} \left(\xi_1 k_3 \xi_2^{-1} + 1\right)^{-1} \|h\|. \tag{5.46}$$

Fazendo $\mathcal{E}h = \overline{h}$ vem

$$\begin{aligned} \|h\| = \left\|\mathcal{E}^{-1}\overline{h}\right\| &> \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)^{-1} \left(\xi_1 k_3 \xi_2^{-1} + 1\right)^{-1} \left\|\overline{h}\right\| \\ &= \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)^{-1} \left(\xi_1 k_3 \xi_2^{-1} + 1\right)^{-1} \|\mathcal{E}h\| \end{aligned}$$

por (5.46). Em particular para $h = \nabla\Phi_k$ tem-se

$$\begin{aligned} \|\nabla\Phi_k\| &> \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)^{-1} \left(\xi_1 k_3 \xi_2^{-1} + 1\right)^{-1} \|\{(1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k\} \nabla\Phi_k\| \\ &= \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)^{-1} \left(\xi_1 k_3 \xi_2^{-1} + 1\right)^{-1} \|d_k\| \end{aligned}$$

donde

$$\|d_k\| < \left(1 + \widetilde{\delta}\gamma_\epsilon \gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)\|\nabla\Phi_k\|,$$

(ou usando o Lema 5.5). De acordo com (5.29), tem-se

$$\|d_k\| < M$$

com $M = \left(1 + \widetilde{\delta}\gamma_\epsilon \gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)\gamma_{\nabla\Phi}\sigma > 0$, sendo $\sigma = máx\,\|x_k - x^*\|$, para $x_k \in L^0$. Prova de iv): usando um raciocínio análogo ao usado para obter (5.44) vem

$$\begin{aligned} -\nabla\Phi_k^T d_k &= d_k^T \left\{(1-w_k)\mathbf{C}_k + w_k \mathbf{B}_k\right\}^{-1} d_k \\ &> (\eta_1 + 2)^{-1}\|d_k\|_2^2. \end{aligned} \tag{5.47}$$

por (5.42). Considere-se $\alpha_k \in (0,1]$ tal que $[x_k, x_k + \alpha_k d_k] \subset L^0$. Pelo teorema do valor médio (Teorema B.3 no Apêndice B ou [28, pp 70]), existe $\widehat{\alpha} \in (0, \alpha_k]$ tal que

$$\Phi(x_k) - \Phi(x_k + \alpha_k d_k) = -\alpha_k \nabla\Phi(x_k)^T d_k - \frac{1}{2}\alpha_k^2 d_k^T \nabla^2\Phi(x_k + \widehat{\alpha} d_k) d_k.$$

De (5.39) e (5.47)

$$\begin{aligned} \Phi(x_k) - \Phi(x_k + \alpha_k d_k) &> \alpha_k (\eta_1 + 2)^{-1}\|d_k\|_2^2 - \frac{1}{2}\alpha_k^2 \gamma_1 \|d_k\|_2^2 \\ &= \alpha_k \left[(\eta_1 + 2)^{-1} - \frac{1}{2}\alpha_k \gamma_1\right]\|d_k\|_2^2 \\ &> \alpha_k \left[(\eta_1 + 2)^{-1} - \frac{1}{2}\gamma_1\right]\|d_k\|_2^2 \\ &> \alpha_k \left[(\eta_1 + 2)^{-1} - \frac{1}{2}\widehat{\gamma}_1\right]\|d_k\|_2^2. \end{aligned}$$

De acordo com o Lema 5.4, $(\eta_1 + 2)^{-1} - \frac{1}{2}\widehat{\gamma}_1 \geq 0$ e de (5.44) tem-se

$$\Phi(x_k) - \Phi(x_k + \alpha_k d_k) > -\alpha_k \left[(\eta_1 + 2)^{-1} - \frac{1}{2}\widehat{\gamma}_1\right](\eta_0 + l_0)\overline{\epsilon}\nabla\Phi_k^T d_k$$

donde

$$\Phi(x_k + \alpha_k d_k) < \Phi(x_k) + \mu\alpha_k \nabla\Phi_k^T d_k$$

em que $\mu = \left[(\eta_1 + 2)^{-1} - \frac{1}{2}\widehat{\gamma}_1\right](\eta_0 + l_0)\overline{\epsilon}$ e por (5.40) e (5.41) satisfaz $0 < \mu < 1$. A condição v) surge naturalmente por i) e por iv) pela aplicação do Lema B.5 no Apêndice B, pois $x_{k+1} = x_k + d_k \subset L^0$, donde, por indução em k, $\{x_k\} \subset L^0$. ■

Teorema 5.3 *Considerem-se as condições do Teorema 5.2. Seja x_0 um ponto inicial da sucessão $\{x_k\}$ definida por $x_{k+1} = x_k + \alpha_k d_k$ com d_k definida em (5.35). Então*

$$\lim_{k\to\infty} \|\nabla\Phi_k\| = 0.$$

A demonstração deste teorema é imediata face às propriedades resultantes do Teorema 5.2 e segue os mesmos passos da demonstração do teorema enunciado no Apêndice B, como Teorema B.7, em [85].

Lema 5.6 *Considerem-se as condições do Teorema 5.3. Então*

$$\lim_{k\to\infty} \|d_k\| = 0.$$

Prova. Considerando os limites inferior e superior de $\|d_k\|$, obtidos respectivamente em (5.45) e (5.37) vem

$$(\eta_0 + l_0)\overline{\epsilon}\,\|\nabla\Phi_k\|_2 \leq \|d_k\|_2 < \left(1 + \widetilde{\delta}\gamma_\epsilon\gamma_{\nabla\Phi}\right)\left(\xi_1 k_3 \xi_2^{-1} + 1\right)(\widehat{\delta})^{-1}\widetilde{\delta}\,\|\nabla\Phi_k\|_2$$

com $0 < \widetilde{\delta} < \infty$ de $\|\nabla\Phi_k\| \leq \widetilde{\delta}\,\|\nabla\Phi_k\|_2$ e $0 < \widehat{\delta} < \infty$ de $\|d_k\|_2 \leq (\widehat{\delta})^{-1}\,\|d_k\|$ (pelo Teorema B.1 no Apêndice B). Como pelo Teorema 5.3 se tem $\lim_{k\to\infty} \|\nabla\Phi_k\| = 0$ então $\lim_{k\to\infty} \|d_k\| = 0$. ■

Face ao Teorema 5.3 e Lema 5.6, é possível enunciar o teorema (idêntico ao Lema B.6 no Apêndice B) que estabelece que a sucessão $\{x_k\}$ converge para o ponto de acumulação x^*, que é ponto estacionário.

Teorema 5.4 *Seja $\Phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ uma função continuamente diferenciável no compacto $L^0 \subset D$ e Ω o conjunto finito dos pontos estacionários de Φ em L^0. Considere-se uma sucessão $\{x_k\} \subset L^0$ definida por $x_{k+1} = x_k + d_k$ com d_k definida em (5.35), para a qual*

$$\lim_{k\to\infty} (x_{k+1} - x_k) = 0 \qquad e \qquad \lim_{k\to\infty} \nabla\Phi_k = 0,$$

respectivamente do Lema 5.6 e do Teorema 5.3. Então

$$\lim_{k\to\infty} x_k = x^* \qquad e \qquad \nabla\Phi(x^*) = 0$$

com $x^* \in \Omega$.

A demonstração deste teorema pode ser consultada em [88, pp 476].

O Teorema 5.5 que irá aparecer adiante é importante porque caracteriza a razão de convergência global do método pseudo-Newton modificado.

Considerem-se ainda as seguintes proposições, retiradas de [28] (veja-se Lema B.1 e Lema B.2 no Apêndice B):

Proposição 5.4 *Seja* $\nabla\Phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ *uma função continuamente diferenciável num conjunto* $L^0 \subset D$. *Considere-se a Hipótese 5.9. Para cada* x_{k+1} *e* $x_k \in L^0$ $(x_{k+1} \neq x_k)$

$$\left\|\nabla\Phi(x_{k+1}) - \nabla\Phi(x_k) - \nabla^2\Phi(x_k)(x_{k+1} - x_k)\right\| \leq \frac{K}{2}\left\|x_{k+1} - x_k\right\|^2 \tag{5.48}$$

em que $x_{k+1} = x_k + d_k$.

A Proposição 5.5 é baseada no Lema 4.1.15 de [28].

Proposição 5.5 *Considere-se* $\nabla\Phi$ *e* $\nabla^2\Phi$ *satisfazendo as condições da Proposição 5.4. Assuma-se que a inversa de* $\nabla^2\Phi$ *existe. Então, para cada* x_{k+1}, x_k *e* $x^* \in L^0 \subset D$ $(x_{k+1} \neq x_k)$ *tem-se*

$$\begin{aligned}&\left\|\nabla\Phi(x_{k+1}) - \nabla\Phi(x_k) - \nabla^2\Phi(x^*)(x_{k+1} - x_k)\right\| \leq \\ &\leq K\text{máx}\{\left\|x_{k+1} - x^*\right\|, \left\|x_k - x^*\right\|\}\left\|x_{k+1} - x_k\right\|.\end{aligned} \tag{5.49}$$

Para a análise que se segue serão precisas as seguintes definições:

Definição 5.5 *Seja* $\{x_k\}$ *uma qualquer sucessão que converge para* x^*. *Dadas duas funções reais de variável real, não negativas e contínuas,* $g(x)$ *e* $h(x)$, *diz-se que*

$$g(x_k) = o(h(x_k))$$

quando $k \to \infty$, *se*

$$\lim_{k\to\infty} \frac{g(x_k)}{h(x_k)} = 0.$$

Definição 5.6 *Seja* $\{x_k\}$ *uma qualquer sucessão que converge para* x^*. *Diz-se que* $x_k \to x^*$ *superlinearmente se*

$$\left\|x_{k+1} - x^*\right\| = o(\left\|x_k - x^*\right\|) \tag{5.50}$$

quando $k \to \infty$.

Para a demonstração do Teorema 5.5 precisamos dos seguintes lemas e proposições: Hipótese 5.10 e Lemas 5.7 e 5.8.

A Hipótese 5.10 estabelece que a matriz $\mathcal{F}_k = (1-w_k)\mathbf{C}_k + w_k\mathbf{B}_k$, usada no sistema das equações do método pseudo-Newton modificado, é uma aproximação à inversa, do inglês *approximate inverse* (de acordo com a definição de Kelley [66, pp 6]), da matriz $\nabla^2\Phi(x^*)$. Esta definição vai permitir a utilização dos Lemas de Banach (Lema B.7 e Teorema B.5, no Apêndice B) na demonstração do Lema 5.8.

O Lema 5.7 estabelece que as sucessões $\|d_k\|$ e $\|x_k - x^*\|$ quando $k \to \infty$, convergem para zero com a mesma razão de convergência.

Finalmente o Lema 5.8 estabelece as condições que garantem que a convergência da direcção pseudo-Newton modificada, d_k, para a direcção Newton, $(d_N)_k$, é mais rápida do que a convergência de d_k para zero.

Hipótese 5.10 *A matriz $\mathcal{F}_k$ definida por*

$$\mathcal{F}_k = (1 - w_k)\mathbf{C}_k + w_k\mathbf{B}_k, \tag{5.51}$$

com $\mathbf{C}_k$ e $\mathbf{B}_k$ definidas respectivamente por (5.17) e (5.18), $w_k \in [0,1)$ e $\{x_k\} \subset L^0$ é uma aproximação à inversa de $\nabla^2\Phi(x^*)$, com

$$\left\| I - \mathcal{F}_k\nabla^2\Phi(x^*) \right\| < 1. \tag{5.52}$$

Nestas condições $\mathcal{F}_k\nabla^2\Phi(x^*)$ é uma matriz não singular (Lema B.7 no Apêndice B). De facto pela Hipótese 5.1, $\nabla^2\Phi^{-1}(x^*)$ existe e de (5.42) (Teorema 5.2) já se provou que $\mathcal{F}_k$ é não singular e d_k, a direcção pseudo-Newton modificada, é bem definida.

Lema 5.7 *Considerem-se satisfeitas as condições do Teorema 5.2 e seja $\{x_k\} \subset L^0$ uma sucessão definida por $x_{k+1} = x_k + d_k$ com d_k definida por (5.35). Então*

$$\widetilde{N}\frac{1}{\gamma_{\nabla\Phi}}\|x_k - x^*\| \leq \|d_k\| \leq 2\,máx\left\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\right\} \tag{5.53}$$

com $\widetilde{N} = (\eta_0 + l_0)\,\overline{\epsilon}\,\widehat{\delta}\left(\widetilde{\delta}\right)^{-1}$ $\quad (0 < \widehat{\delta},\ \widetilde{\delta} < \infty)$.

Prova. Usando (5.29) em (5.45) tem-se

$$\begin{aligned}
\|d_k\| &\geq \widehat{\delta}\,\|d_k\|_2 \geq \widehat{\delta}(\eta_0 + l_0)\,\overline{\epsilon}\,\|\nabla\Phi_k\|_2 \\
&\geq \widehat{\delta}(\eta_0 + l_0)\,\overline{\epsilon}\left(\widetilde{\delta}\right)^{-1}\|\nabla\Phi_k\| \\
&\geq \widetilde{N}\frac{1}{\gamma_{\nabla\Phi}}\|x_k - x^*\|,
\end{aligned}$$

com $\widetilde{N} = N\widehat{\delta}\left(\widetilde{\delta}\right)^{-1}$, $N = (\eta_0 + l_0)\,\overline{\epsilon}$,

$$\|d_k\| \geq \widehat{\delta}\,\|d_k\|_2 \quad \text{e} \quad \|\nabla\Phi_k\|_2 \geq \left(\widetilde{\delta}\right)^{-1}\|\nabla\Phi_k\|\,.$$

Da mesma forma, como

$$\begin{aligned} 0 \;&\leq\; \|d_k\| \leq \|x_{k+1} - x^*\| + \|x_k - x^*\| \\ &\leq\; \|d_k\| \leq 2\,máx\,\{\|x_{k+1} - x^*\|\,, \|x_k - x^*\|\}\,. \end{aligned}$$

■

Lema 5.8 *Considerem-se as condições do Teorema 5.2 e a Hipótese 5.9. Seja* $\{x_k\} \subset L^0$ *uma sucessão definida por* $x_{k+1} = x_k + d_k$ *com* d_k *definida por (5.35) e* $\mathcal{F}_k$, *definida por (5.51), uma aproximação à inversa de* $\nabla^2\Phi(x^*)$. *Então*

$$\|d_k - (d_N)_k\| = o(\|d_k\|)$$

em que $(d_N)_k$ *é a direcção Newton calculada na iteração* k.

Prova. Considere-se a seguinte norma:

$$\|d_k - (d_N)_k\| = \left\|\left(\mathcal{F}_k - \nabla^2\Phi_k^{-1}\right)\nabla\Phi_k\right\|$$

com $\mathcal{F}_k$ definida por (5.51). Assim,

$$\begin{aligned} \|d_k - (d_N)_k\| \;&=\; \left\|\left(\mathcal{F}_k\nabla^2\Phi_k - I\right)\nabla^2\Phi_k^{-1}\nabla\Phi_k\right\| \qquad (5.54) \\ &=\; \left\|\left\{\mathcal{F}_k\left(\nabla^2\Phi_k - \nabla^2\Phi(x^*)\right) + \left(\mathcal{F}_k\nabla^2\Phi(x^*) - I\right)\right\}\nabla^2\Phi_k^{-1}\nabla\Phi\right\| \\ &\leq\; \left\{\|\mathcal{F}_k\|\,K\,\|x_k - x^*\| + \left\|\mathcal{F}_k\nabla^2\Phi(x^*) - I\right\|\right\} k_2\beta\gamma_{\nabla\Phi}\,\|x_k - x^*\| \qquad (5.55) \end{aligned}$$

tendo em conta (5.30), (5.27) e (5.29). De acordo com a Hipótese 5.10 e usando o Lema B.7 no Apêndice B com $M = I - \mathcal{F}_k\nabla^2\Phi(x^*)$

$$\left\|\left(\mathcal{F}_k\nabla^2\Phi(x^*)\right)^{-1}\right\| \leq \frac{1}{1 - \left\|I - \mathcal{F}_k\nabla^2\Phi(x^*)\right\|}$$

donde

$$1 - \left\|\mathcal{F}_k\nabla^2\Phi(x^*)\right\| \leq \left\|I - \mathcal{F}_k\nabla^2\Phi(x^*)\right\|$$

ou

$$1 - \|\mathcal{F}_k\| \left\|\nabla^2\Phi(x^*)\right\| \leq \left\|I - \mathcal{F}_k\nabla^2\Phi(x^*)\right\|. \tag{5.56}$$

Por outro lado, de (5.49), tem-se

$$\|\nabla\Phi_{k+1} - \nabla\Phi_k\| - \left\|\nabla^2\Phi(x^*)\right\| \|d_k\| \leq K máx\left\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\right\} \|d_k\|$$

que origina

$$-K\sigma_k \|d_k\| \leq \left\|\nabla^2\Phi(x^*)\right\| \|d_k\|$$

com $\sigma_k = máx\left\{\|x_{k+1} - x^*\|, \|x_k - x^*\|\right\}$. Dividindo por $\|d_k\|$, esta desigualdade define um limite inferior para a norma de $\nabla^2\Phi(x)$, no ponto x^*, em função de σ_k:

$$\left\|\nabla^2\Phi(x^*)\right\| \geq -K\sigma_k,$$

ou ainda

$$\left\|\nabla^2\Phi(x^*)\right\| \leq k_4 K\sigma_k, \tag{5.57}$$

para algum $k_4 \geq 1$. Da mesma forma, de (5.56) tem-se

$$\begin{array}{rcl} \left\|I - \mathcal{F}_k\nabla^2\Phi(x^*)\right\| & \geq & -\|\mathcal{F}_k\| \left\|\nabla^2\Phi(x^*)\right\| \\ \left\|I - \mathcal{F}_k\nabla^2\Phi(x^*)\right\| & \leq & k_5 \|\mathcal{F}_k\| \left\|\nabla^2\Phi(x^*)\right\|, \end{array} \tag{5.58}$$

para algum $k_5 \geq 1$. Assim, considerando (5.55) e usando sucessivamente (5.58) e (5.57), obtém-se

$$\begin{array}{rcl} \|d_k - (d_N)_k\| & \leq & \left\{\|\mathcal{F}_k\| K \|x_k - x^*\| + k_4k_5 \|\mathcal{F}_k\| K\sigma_k\right\} k_2\beta\gamma_{\nabla\Phi} \|x_k - x^*\| \\ & \leq & \left(\|x_k - x^*\| + k_4k_5\sigma_k\right) \|\mathcal{F}_k\| K k_2\beta\gamma_{\nabla\Phi} \|x_k - x^*\| \\ & = & \widetilde{M} \|x_k - x^*\| \\ & = & o(\|x_k - x^*\|), \end{array}$$

com $\widetilde{M} = \left(\|x_k - x^*\| + k_4k_5\sigma_k\right) \|\mathcal{F}_k\| K k_2\beta\gamma_{\nabla\Phi}$, pois $\widetilde{M} \to 0$ quando $k \to \infty$, pelo Lema 5.4 e para toda a sucessão $\{x_k\} \subset L^0$. Devido a (5.53) (Lema 5.7), tem-se

$$\|d_k - (d_N)_k\| = o(\|x_k - x^*\|) = o(\|d_k\|).$$

■

Enuncia-se de seguida um lema bem conhecido:

Lema 5.9 *(Lema 8.23 de [28]) Seja* $\{x_k\} \subset L^0$ *uma sucessão que converge superlinearmente para* x^*. *Então*

$$\lim_{k\to\infty} \frac{\|d_k\|}{\|x_k - x^*\|} = 1.$$

A propriedade enunciada no Lema 5.9 vai ser usada na demonstração do Teorema 5.5. O enunciado deste teorema e respectiva prova usam princípios, de certo modo, semelhantes aos usados nos teoremas de convergência dos métodos tipo Quasi-Newton ([85, pp 304] e [28, pp 181]).

Teorema 5.5 *Seja* $\Phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ *uma função real de variável real continuamente diferenciável no compacto* $L^0 \subset D$. *Considerem-se satisfeitas as condições do Teorema 5.2 e do Lema 5.8. Seja* $\{x_k\} \subset L^0$ *uma sucessão calculada por* $x_{k+1} = x_k + d_k$ *com* d_k *definida por (5.35) onde* $\{\mathcal{F}_k\} \in \mathbb{R}^{n\times n}$ *é uma sucessão de matrizes não singulares (Teorema 5.2) em que*

$$\lim_{k\to\infty} x_k = x^* \in L^0$$

$(x_k \neq x^*)$ *para todo o* k *e* $\nabla\Phi(x^*) = 0$. *Suponha-se também que* $\nabla^2\Phi(x^*)$ *é não singular. Então* $\{x_k\}$ *converge para* x^* *superlinearmente se e só se*

$$\|d_k - (d_N)_k\| = o(\|d_k\|) \tag{5.59}$$

em que $(d_N)_k$ *é a direcção Newton calculada em* x_k.

Prova. Considere-se, primeiro, que a condição (5.59) se verifica. Como

$$\begin{aligned}
\nabla^2\Phi_k\left(d_k - (d_N)_k\right) &= \nabla^2\Phi_k d_k + \nabla\Phi_k \\
&= \nabla^2\Phi_k d_k + \nabla\Phi_k - \nabla\Phi_{k+1} + \nabla\Phi_{k+1} \\
&= \nabla\Phi_{k+1} - (\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k d_k)
\end{aligned}$$

tomando normas

$$\begin{aligned}
\left\|\nabla^2\Phi_k\left(d_k - (d_N)_k\right)\right\| &\geq \|\nabla\Phi_{k+1}\| - \left\|\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k d_k\right\| \\
&\geq \|\nabla\Phi_{k+1}\| - \frac{K}{2}\|d_k\|^2
\end{aligned} \tag{5.60}$$

pela Proposição 5.4. Além disso, como

$$\left\|\nabla^2\Phi_k\left(d_k - (d_N)_k\right)\right\| \leq \left\|\nabla^2\Phi_k\right\| \|d_k - (d_N)_k\|$$

tem-se

$$\lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k\left(d_k-(d_N)_k\right)\right\|}{\|d_k\|} \leq \lim_{k\to\infty} \left\|\nabla^2\Phi_k\right\| \frac{\|d_k-(d_N)_k\|}{\|d_k\|} = 0$$

por hipótese (condição (5.59)) e pelo facto de $\left\|\nabla^2\Phi(x_k)\right\|$ ter um limite finito em virtude da Hipótese 5.9 de continuidade. Assim

$$\begin{aligned} 0 &\geq \lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k\left(d_k-(d_N)_k\right)\right\|}{\|d_k\|} \\ &\geq \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} - \frac{K}{2} \lim_{k\to\infty} \|d_k\| \end{aligned}$$

por (5.60), e

$$\lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} = 0, \tag{5.61}$$

pelo Lema 5.6. Como por hipótese $\nabla\Phi(x^*)=0$ e de (5.29)

$$\|\nabla\Phi_{k+1}\| \geq \frac{1}{\gamma_{\nabla\Phi}} \|x_{k+1}-x^*\|$$

e

$$\begin{aligned} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} &\geq \frac{\frac{1}{\gamma_{\nabla\Phi}}\|x_{k+1}-x^*\|}{\|x_{k+1}-x^*\|+\|x_k-x^*\|} \\ &= \frac{\rho_k}{\gamma_{\nabla\Phi}(\rho_k+1)} \end{aligned}$$

com $\rho_k = \dfrac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}$. De (5.61) conclui-se que

$$\lim_{k\to\infty} \frac{\rho_k}{\gamma_{\nabla\Phi}(\rho_k+1)} = 0$$

e $\{\rho_k\}$ converge para zero, o que prova a convergência q-superlinear da sucessão $\{x_k\}$. Assuma-se agora que $\{x_k\}$ converge q-superlinearmente para x^*. Como

$$\frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} = \frac{\|\nabla\Phi_{k+1}-\nabla\Phi(x^*)\|\,\|x_k-x^*\|}{\|x_k-x^*\|\,\|d_k\|}$$

por (5.29) tem-se

$$\begin{aligned} \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} &\leq \lim_{k\to\infty} \frac{\gamma_{\nabla\Phi}\,\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \lim_{k\to\infty} \frac{\|x_k-x^*\|}{\|d_k\|} \\ &= 0 \end{aligned} \tag{5.62}$$

por hipótese e pelo Lema 5.9. Como

$$-\nabla\Phi_{k+1} = -(\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k d_k) - \nabla\Phi_k - \nabla^2\Phi_k d_k$$

tomando normas

$$\begin{aligned} \|\nabla\Phi_{k+1}\| &\geq -\left\|\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k d_k\right\| + \left\|-\nabla\Phi_k - \nabla^2\Phi_k d_k\right\| \\ &\geq \left\|-\nabla\Phi_k - \nabla^2\Phi_k d_k\right\| - \frac{K}{2}\|d_k\|^2 \end{aligned} \tag{5.63}$$

pela Proposição 5.4. Além disso

$$\begin{aligned} \left\|-\nabla\Phi_k - \nabla^2\Phi_k d_k\right\| &= \left\|\nabla^2\Phi_k (d_N)_k - \nabla^2\Phi_k d_k\right\| \\ &= \left\|\nabla^2\Phi_k ((d_N)_k - d_k)\right\|. \end{aligned}$$

De (5.63) obtém-se

$$\frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} \geq \frac{\left\|\nabla^2\Phi_k (d_k - (d_N)_k)\right\|}{\|d_k\|} - \frac{K}{2}\|d_k\|,$$

donde

$$\lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k (d_k - (d_N)_k)\right\|}{\|d_k\|} = 0$$

por (5.62) e pelo Lema 5.6. A condição (5.59) surge naturalmente pela equivalência das normas num espaço vectorial de dimensão finita, isto é, existe uma constante $\gamma > 0$ tal que

$$\|A\| \leq \gamma \|A\|_M$$

para qualquer matriz de ordem $n \times n$ ou vector de $\mathbb{R}^n$. Em particular, se $M = \nabla^2\Phi_k$ não singular (em L^0), então

$$\|d_k - (d_N)_k\|_M = \|M\,(d_k - (d_N)_k)\|$$

([72, pp 105]) define uma norma para vectores de $\mathbb{R}^n$. Logo

$$\lim_{k\to\infty} \frac{\|d_k - (d_N)_k\|_{\nabla^2\Phi_k}}{\|d_k\|} = \lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k\,(d_k - (d_N)_k)\right\|}{\|d_k\|} = 0$$

como se pretendia. ■

Tendo provado na condição i) do Teorema 5.2 que

$$\nabla\Phi_k^T d_k < -\varphi\,\|\nabla\Phi_k\|\,\|d_k\|\,, \quad \varphi > 0$$

e de (5.45), verifica-se que

$$\nabla\Phi_k^T d_k < -\overline{\varphi}\,\|d_k\|^2\,, \quad \overline{\varphi} = \varphi\frac{1}{(\eta_0 + l_0)\,\overline{\epsilon}} > 0$$

para a função de penalidade $\Phi(x, r)$ (de (4.12)) e para a direcção em estudo - pseudo-Newton modificada, d_{PNM_k}- calculada a partir da equação (5.23) em função de $\nabla\Phi_k$. Nestas condições é possível utilizar o seguinte lema que estabelece que o efeito de Maratos não ocorre e a razão de convergência superlinear do algoritmo não é destruída. Este lema retirado do trabalho de Facchinei e Lucidi [36] (Lema B.8 no Apêndice B), foi adaptado ao caso em estudo.

Lema 5.10 *Considerem-se as condições do Teorema 5.2. Então existe $\overline{\varphi} > 0$ e um conjunto compacto L^0, tal que, para todo o $x_k \in L^0$ se tem*

$$\nabla\Phi(x_k)^T d_k < -\overline{\varphi}\,\|d_k\|^2$$

e para todo o $\mu \in (0, \frac{1}{2})$

$$\Phi(x_k + d_k) - \Phi(x_k) \leq \mu\nabla\Phi(x_k)^T d_k.$$

5.8 Síntese e comentários

A resolução do problema de optimização não linear com restrições de igualdade foi feita numa perspectiva de penalidade da função objectivo do problema de acordo com o grau de violação das restrições. A abordagem seleccionada origina uma técnica de penalidade exacta, com vantagens significativas no que concerne à escolha e ajuste do parâmetro de penalidade. No entanto, a complexidade das fórmulas das derivadas da função de penalidade resultante da estratégia implementada é notória e desmotivadora.

O passo seguinte estaria obviamente relacionado com a utilização de expressões mais simples que aproximem as derivadas da função de penalidade, reduzam o custo computacional envolvido nos cálculos e não afectem o desempenho do algoritmo final. As pseudo-derivadas introduzidas neste capítulo originam aproximações equivalentes a outras técnicas já existentes só que obtidas segundo uma perspectiva de técnicas de penalidade. O estudo da convergência não trouxe surpresas. Uma convergência superlinear de acordo com certas condições já era esperada, tendo em conta que o método não recorre às derivadas de ordem dois e superior das funções de restrição e de ordem três e superior da função objectivo. O facto da convergência ser global deve-se em parte à introdução da estratégia relacionada com a função $w(x)$, que gera as direcções pseudo-Newton modificadas, que equilibram, quando necessário, o peso das direcções pseudo-Newton e pseudo-descida máxima. A direcção pseudo-Newton modificada satisfaz assim algumas propriedades, designadamente, é suficientemente descendente e é do tipo gradiente. Paralelamente, a introdução da condição de Armijo e de um esquema de sucessivas reduções do comprimento do passo, gera um algoritmo globalmente convergente.

A procura de aproximações à solução, ao longo da direcção pseudo-Newton modificada, originou resultados promissores e o desempenho do algoritmo pode considerar-se satisfatório.

As experiências computacionais realizadas mostraram mesmo assim que um algoritmo baseado numa técnica de penalidade nem sempre gera aproximações suficientemente perto da região admissível, excepto quando já perto da solução. Este facto conduziu à implementação de uma estratégia que obrigasse o iterando a aproximar-se da região admissível através do cálculo de um vector correcção. O passo final seria assim constituído por duas componentes. Um vector que dirigisse o iterando na direcção do ponto estacionário da função de penalidade e outro que o encaminhasse no sentido da região admissível. Esta estratégia é o tema do próximo capítulo designado por procura curvilínea.

Capítulo 6

Procura curvilínea

Na grande maioria dos trabalhos sobre a procura curvilínea, a estratégia implementada consiste em adicionar à direcção de procura um vector de curvatura negativa, de forma a acelerar a convergência do processo iterativo. Neste capítulo vai apresentar-se uma estratégia com o mesmo objectivo, acelerar a convergência, mas com uma filosofia diferente. Assim, a correcção a adicionar tem como finalidade forçar o processo iterativo na direcção da região admissível. É nesta perspectiva que surge a procura curvilínea pela adição de um vector correcção à direcção de procura pseudo-Newton modificada, de forma a que a aproximação seguinte verifique as restrições. A procura é, desta forma, efectuada ao longo de um arco.

Começa-se por definir o vector correcção e o algoritmo de procura unidimensional para o cálculo do comprimento do passo. A seguir faz-se uma análise da razão de convergência.

6.1 O estado da arte

A estratégia usual de procura num processo iterativo usa apenas uma direcção, ficando a selecção do próximo ponto a fazer-se ao longo da linha definida por essa mesma direcção. Este capítulo apresenta uma filosofia diferente, na qual em cada iteração a procura pode ser efectuada ao longo de um arco, construído com base num modelo quadrático que usa duas direcções calculadas em cada iteração. Na iteração k, se as duas direcções calculadas forem representadas pelos vectores d_k e $\widetilde{d}_k$, o próximo ponto do processo iterativo pode ser calculado através do modelo quadrático em α definido por

$$x(\alpha) = x_k + \alpha d_k + \alpha^2 \widetilde{d}_k, \quad \alpha \in \mathbb{R}.$$

A figura seguinte pretende ilustrar, o processo de procura curvilínea na iteração k. Os pontos do gráfico representam as aproximações para diferentes valores de $\alpha \in (0, 1]$,

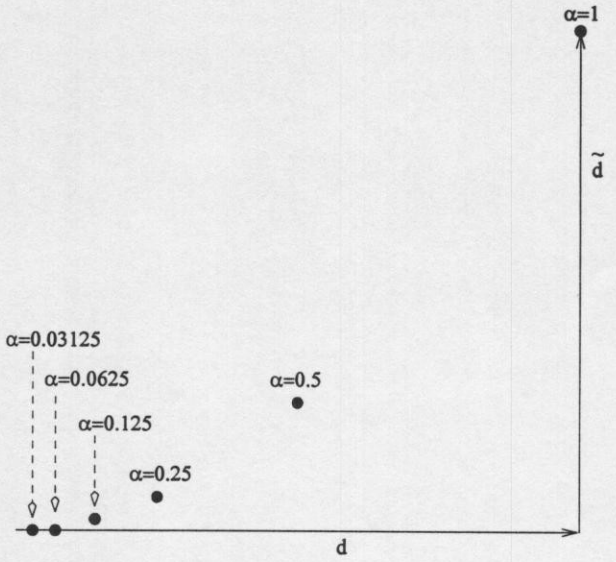

Figura 6.1: Procura curvilínea

definindo uma curva. Por uma questão de simplificação do desenho, as duas direcções d_k e $\tilde{d}_k$ são colocadas perpendicularmente, o que pode não corresponder à realidade.

A procura curvilínea tem vindo a ser estudada há já algum tempo por alguns autores no âmbito da optimização com e sem restrições. No contexto da optimização sem restrições, são de destacar os seguintes trabalhos: [75], [83], [55], [15], [70], [71], [40], [59] e [34]. Para todos eles, e de uma forma ligeira, vai ser feito um resumo das ideias gerais.

McCormick em [75] apresenta uma modificação do procedimento do cálculo do passo de procura baseado no critério de Armijo, de forma a incluir informação das segundas derivadas, usada em particular, quando a matriz Hessiana não é semi-definida positiva. Mostra como uma modificação do critério de Armijo pode ser usada com direcções de curvatura negativa. A procura define uma sucessão de valores da função monótona decrescente e o termo de correcção é uma direcção não ascendente e de curvatura não positiva. Moré e Sorensen em [83] fazem uma extensão do trabalho de McCormick e apresentam duas formas para o cálculo de direcções de curvatura negativa - uma baseada na factorização modificada de Cholesky e outra baseada na factorização de Bunch e Parlett. Também Goldfarb em [55], propõe modificações às regras de Armijo e Goldstein para o cálculo do comprimento do passo, incorporando-lhes informação acerca das segundas derivadas, por forma a permitir passos ao longo dum percurso curvilíneo. A direcção descendente

proposta é a simétrica do gradiente e o termo de correcção é uma direcção não ascendente e de curvatura negativa, baseada no cálculo de valores próprios da matriz Hessiana.

Conforti, Grandinetti e Musmanno em [15] propõem uma noção de trajectória curvilínea e descrevem algumas das suas propriedades, no contexto da optimização não linear contínua sem restrições. Sugerem uma trajectória curvilínea definida por um tensor das derivadas parciais de terceira ordem da função objectivo. Uma trajectória $x(\alpha)$ é definida como um espaço unidimensional não linear parametrizado pela constante $\alpha \in \mathbb{R}^+$. Na implementação, é feita uma aproximação quadrática da trajectória curvilínea, através da introdução de tensores simétricos da função objectivo em cada aproximação à solução. Devido à complexidade da estrutura computacional, é sugerida a utilização de diferenciação automática e de computação paralela.

Ainda nos problemas sem restrições mas de grande dimensão, Lucidi, Rochetich e Roma em [70] apresentam um algoritmo não monótono de procura curvilínea cujas direcções são calculadas através do método de Lanczos. Uma direcção é do tipo Newton e a outra é de curvatura negativa, que tem em conta a não convexidade da função objectivo. Com este trabalho pretendem estudar a conjugação da curvatura negativa com uma estratégia não monótona. No seguimento deste trabalho e usando a mesma ideia, Lucidi e Roma em [71] introduzem algoritmos de Newton truncados bastante eficientes.

Ferris, Lucidi e Roma em [40] combinam uma aproximação da direcção Newton com uma de curvatura negativa, ambas calculadas usando a factorização Bunch-Parlett, numa estratégia de procura curvilínea não monótona. Também Gould, Lucidi, Roma e Toint em [59] apresentam direcções de curvatura negativa no âmbito da optimização sem restrições.

Erikson em [34] propõe uma procura curvilínea muito fácil de implementar que deve ser usada com o método Gauss-Newton, para a resolução de sistemas de equações não lineares. A procura curvilínea pode também ser utilizada para problemas de mínimos quadrados não lineares que originam sistemas sobredeterminados.

Passando para a optimização com restrições, são referências os trabalhos [110], [68] e [91]. Zhu em [110] apresenta um algoritmo para restrições de igualdade no qual se usa o método Quasi-Newton projectado e uma estratégia de procura unidimensional não monótona. Uma direcção é calculada a partir da projecção à esquerda e à direita da aproximação à Hessiana. O termo de correcção tem como objectivo principal evitar o efeito de Maratos. Liao em [68] estuda problemas com restrições genéricas. A estratégia baseia-se no método da Hessiana reduzida e, tal como Zhu em [110], o termo de correcção é usado para combater o efeito de Maratos e assim proteger a razão de convergência superlinear. Neste trabalho o termo de correcção é de segunda ordem e baseia-se nas restrições activas. Problemas com restrições do tipo desigualdade são estudados por Panier e Tits em [91]. Estes autores apresentam um algoritmo baseado na programação quadrática sequencial com utilização de actualizações Quasi-Newton para aproximar a Hessiana da

função Lagrangeana associada ao problema. A procura é monótona decrescente e exige admissibilidade em todas as iterações. A exigência de monotonia e admissibilidade podem não permitir o passo unitário na vizinhança da solução, comprometendo a razão de convergência superlinear. Estas dificuldades são ultrapassadas pelos autores, recorrendo a um encurvamento da direcção de procura, sendo o termo de correcção a solução de outro problema quadrático.

Para além dos trabalhos [68] e [110], também Panier e Tits em [90], sugerem a procura curvilínea como uma solução alternativa aos esquemas de vigilância para ultrapassar o efeito de Maratos ([14]).

6.2 Vector correcção

O método pseudo-Newton modificado baseia-se numa técnica de penalidade exacta. Desta forma, ao longo do processo iterativo, as várias aproximações geradas podem ou não pertencer à região admissível. Esta característica das técnicas de penalidade torna-as preferenciais para os problemas com restrições do tipo igualdade.

Com o propósito de acelerar a convergência do processo, pretende-se encaminhar as várias aproximações na direcção da região admissível. Assim, se o próximo iterando $x_{k+1} = x_k + d_{PNM_k}$ não verificar as restrições $c_i(x)$, $i = 1, \ldots, m$, calcula-se um vector correcção, $\widetilde{d}_k$, por forma a que as restrições sejam verificadas, ou seja, $c(x_k + d_{PNM_k} + \widetilde{d}_k) = 0$. Na prática, a verificação da admissibilidade para um ponto x_k traduz-se por $\|c(x_k)\| \leq \delta_2$, para δ_2 pequeno e positivo.

O sistema de m equações não lineares nas n incógnitas

$$c(x_k + d_{PNM_k} + \widetilde{d}_k) = 0$$

resolve-se utilizando o método de Newton, dando origem ao sistema linear indeterminado

$$A(x_k + d_{PNM_k})\widetilde{d}_k = -c(x_k + d_{PNM_k}), \tag{6.1}$$

com $A(x) = \nabla c(x)^T$.

Na explicação que se segue as matrizes ∇c, A, Q, R, L, Y e Z , são calculadas para um certo valor x, mas por uma questão de simplificação da notação, em vez de, por exemplo $Q(x)$, vai usar-se apenas Q.

A matriz ∇c é factorizada através da factorização QR baseada no método de Householder

$$AQ = R^T, \quad \text{com } Q = [Y|Z], \quad R^T = [L|O]$$

em que $Q \in \mathbb{R}^{n\times n}$ é uma matriz ortogonal formada por $Y \in \mathbb{R}^{n\times m}$, matriz das primeiras m colunas de Q e $Z \in \mathbb{R}^{n\times(n-m)}$. O factor $R^T \in \mathbb{R}^{m\times n}$ é composto pela matriz triangular inferior $L \in \mathbb{R}^{m\times m}$ e pela matriz nula $O \in \mathbb{R}^{m\times(n-m)}$.

Como a matriz Q é ortogonal tem-se

$$AQ = R^T \Leftrightarrow AQQ^T = R^TQ^T \Leftrightarrow A = (QR)^T \Leftrightarrow \nabla c = QR,$$

apresentando o produto R^TQ^T a forma

$$R^TQ^T = \left[\begin{array}{c|c} L & O \end{array}\right] \left[\begin{array}{c} Y^T \\ Z^T \end{array}\right] = LY^T + OZ^T.$$

A forma factorizada para A fica

$$A = LY^T.$$

Voltando ao sistema (6.1) e usando a factorização de A tem-se

$$L(x_k + d_{PNM_k})Y(x_k + d_{PNM_k})^T\widetilde{d}_k = -c(x_k + d_{PNM_k}) \tag{6.2}$$

que é um sistema com menos equações do que incógnitas. Para verificar a equação (6.2), o vector $\widetilde{d}_k$ é calculado a partir de

$$\widetilde{d}_k = Y(x_k + d_{PNM_k})s_k \tag{6.3}$$

com

$$L(x_k + d_{PNM_k})s_k = -c(x_k + d_{PNM_k})$$

e $s_k \in \mathbb{R}^m$.

6.3 Procura unidimensional

Em cada iteração, a procura do próximo iterando é feita ao longo de uma curva, quadrática, em α_k

$$x_{k+1} = x_k + \alpha_k d_{PNM_k} + \alpha_k^2\widetilde{d}_k \tag{6.4}$$

com base nas direcções d_{PNM_k} e $\widetilde{d}_k$ calculadas para a iteração k.

Como $\alpha_k \in (0, 1]$, o facto de d_{PNM_k} aparecer multiplicado por α_k e $\widetilde{d}_k$ por α_k^2 justifica-se pois d_{PNM_k} é uma direcção que se dirige para um ponto estacionário de Φ e deve ter uma contribuição mais relevante para a direcção resultante do que $\widetilde{d}_k$.

O algoritmo que se segue pretende retratar a filosofia da procura curvilínea:

Algoritmo 6.1 (Método pseudo-Newton modificado com procura curvilínea)

Dados $\overline{\eta}_1$, $\overline{\mu}_1$, δ_1, δ_2 e x_0;
Enquanto *(Critério de paragem = falso)* ***Fazer***
  Se $(D^2\Phi_k$ *é singular)* ***Então*** $w_k = 1$;
  Senão $\{$*calcular* d_{PN_k}, *solução de* $D^2\Phi_k d_{PN_k} = -D\Phi_k\}$;
    Se $(D\Phi_k^T d_{PN_k} \leq -\overline{\eta}_1 \|D\Phi_k\| \|d_{PN_k}\|)$ ***Então*** $w_k = 0$;
    Senão $\{$*calcular* w_k *usando (5.20)*$\}$
    FimSe
  FimSe
  $\{$*Calcular* d_{PNM_k} *usando (5.19)*$\}$
  Se $(\|c(x_k + d_{PNM_k})\| \leq \delta_2)$ ***Então***
    $\{\alpha = 1;\ x = x_k + \alpha d_{PNM_k}\}$
    Enquanto $\left(\Phi(x_k) - \Phi(x) < -\overline{\mu}_1 \alpha D\Phi_k^T d_{PNM_k}\right)$ ***Fazer***
      Se $(\|\alpha d_{PNM_k}\|_2 > \delta_1)$ ***Então***
        $\left\{\alpha = \dfrac{\alpha}{2};\quad x = x_k + \alpha d_{PNM_k}\right\}$
      Senão $\{\alpha = 1;\quad x = x_k + \alpha d_{PNM_k}\}\,;\{SAIR\}$
      FimSe
    FimEnquanto
  $\{x_{k+1} = x\}$
  Senão
    $\{\widetilde{d}_k = -Y(x_k + d_{PNM_k})L^{-1}(x_k + d_{PNM_k})c(x_k + d_{PNM_k})\}$
    $\left\{\alpha = 1;\quad x = x_k + \alpha d_{PNM_k} + \alpha^2 \widetilde{d}_k\right\}$
    Enquanto $\left(\Phi(x_k) - \Phi(x) < -\overline{\mu}_1 \alpha D\Phi_k^T (d_{PNM_k} + \alpha \widetilde{d}_k)\right)$ ***Fazer***
      Se $\left(\left\|\alpha d_{PNM_k} + \alpha^2 \widetilde{d}_k\right\|_2 > \delta_1\right)$ ***Então***$\left\{\alpha = \dfrac{\alpha}{2};\quad x = x_k + \alpha d_{PNM_k} + \alpha^2 \widetilde{d}_k\right\}$
      Senão$\left\{\alpha = 1;\quad x = x_k + \alpha d_{PNM_k} + \alpha^2 \widetilde{d}_k\right\}\,;\{SAIR\}$
      FimSe
    FimEnquanto
    $\{x_{k+1} = x\}$
  FimSe
FimEnquanto

A condição de Armijo modificada que garante uma redução significativa da função, em cada iteração, é a seguinte:

$$\Phi(x_k) - \Phi(x_{k+1}) \geq -\overline{\mu}_1 \alpha_k (d_{PNM_k} + \alpha_k \widetilde{d}_k)^T D\Phi(x_k), \qquad 0 < \overline{\mu}_1 < 1. \tag{6.5}$$

Também foi implementada uma técnica de redução sucessiva de valores de α.

O valor de α_k quando enquadrado na procura curvilínea, pode também ser calculado pelo algoritmo PCCAA (Preditor-Corrector-Curry-Altman-Armijo) ([80]). Apresenta-se de seguida, apenas o correspondente passo do algoritmo. Este é baseado no cálculo de um ponto auxiliar, $x_k + d_{PNM_k} + \widetilde{d}_k$, que corresponde a fazer $\alpha_k = 1$ (valor previsto), a partir do qual se calcula outro valor de α_k (valor corrigido). Destes dois, o que originar menor valor da função Φ, vai verificar o critério de Armijo modificado introduzindo-se uma estratégia de redução sucessiva dos valores de α se necessário.

Algoritmo 6.2 (PCCAA)

Senão

$\{$*calcular* $\widetilde{d}_k$:

$\widetilde{d}_k = -Y(x_k + d_{PNM_k})L^{-1}(x_k + d_{PNM_k})c(x_k + d_{PNM_k})\}$

$\{$*calcular* $D\Phi(x_k + d_{PNM_k} + \widetilde{d}_k)\}$

$$\{\text{calcular } \alpha = \frac{(\eta_2 - 1)D\Phi_k{}^T\left(d_{PNM_k} + \widetilde{d}_k\right)}{\left(D\Phi\left(x_k + d_{PNM_k} + \widetilde{d}_k\right) - D\Phi_k\right)^T\left(d_{PNM_k} + \widetilde{d}_k\right)}\}$$

Se $\Phi(x_k + d_{PNM_k} + \widetilde{d}_k) \leq \Phi(x_k + \alpha d_{PNM_k} + \alpha^2\widetilde{d}_k)$ ***Então***

$\{\alpha = 1\}$

FimSe

$\{x = x + \alpha d_{PNM_k} + \alpha^2\widetilde{d}_k\}$

Enquanto $\Phi(x_k) - \Phi(x) < -\overline{\mu}_1\alpha D\Phi_k{}^T\left(d_{PNM_k} + \alpha\widetilde{d}_k\right)\}$***Fazer***

Se $\left(\left\|\alpha d_{PNM_k} + \alpha^2\widetilde{d}_k\right\|_2 > \delta_1\right)$ ***Então***

$\left\{\alpha = \frac{\alpha}{2}; \quad x = x_k + \alpha d_{PNM_k} + \alpha^2\widetilde{d}_k\right\}$

Senão

$\left\{\alpha = 1; \quad x = x_k + \alpha d_{PNM_k} + \alpha^2\widetilde{d}_k\right\}$

$\{SAIR\}$

FimSe

FimEnquanto

$\{x_{k+1} = x\}$

FimSe

6.4 Estudo da convergência global

Os resultados obtidos das experiências computacionais realizadas com o Algoritmo 6.1, considerando a estratégia de procura unidimensional apresentada no Algoritmo 6.2, foram encorajadores quando comparados com os resultados obtidos com a procura linear do método pseudo-Newton modificado. Este facto, serviu por si só de estímulo para um estudo mais aprofundado do comportamento deste processo iterativo em termos de razão de convergência.

Nesta secção retoma-se a notação d_k para denotar a direcção pseudo-Newton modificada, d_{PNM_k}. Considere-se a sucessão $\{x_k\}$ definida por

$$x_{k+1} = x_k + d_k + \widetilde{d}_k \tag{6.6}$$

cujo vector correcção é definido por

$$\widetilde{d}_k = Y(x_k + d_k)s_k \quad \text{com} \quad L(x_k + d_k)s_k = -c(x_k + d_k). \tag{6.7}$$

Considerem-se as seguintes hipóteses:

Hipótese 6.1 *O operador $A(x) : D \subset \mathbb{R}^n \to \mathbb{R}^{m\times n}$ é contínuo à Lipschitz em x^*, de constante γ_A $(0 < \gamma_A < \infty)$, isto é,*

$$\|A(x_k) - A(x^*)\|_F \leq \gamma_A \|x_k - x^*\|$$

para x_k e $x^* \in L^0$.

Hipótese 6.2 *A matriz $A(x_k)$ tem característica m para todo o x_k no conjunto $L^0 \subset D$.*

Além disso, como consequência desta hipótese a matriz $L(x)$ da factorização de $A(x)$ $(A(x) = L(x)Y(x)^T)$ é não singular.

Hipótese 6.3 *Existem constantes positivas r_1 e r_2 tais que*

$$\|L(x_k)\| \leq r_1 \qquad e \qquad \left\|L^{-1}(x_k)\right\| \leq r_2$$

para todo o $x_k \in L^0$.

De $A(x) = L(x)Y(x)^T$ vem

$$\|A(x^*)\|_F \leq \widetilde{\gamma} \|L(x^*)\| \leq \widetilde{\gamma} r_1 \tag{6.8}$$

pela Hipótese 6.3 e para $0 < \widetilde{\gamma} < \infty$ (Teorema B.1 no Apêndice B).

Proposição 6.1 *Seja $c(x) : D \subset \mathbb{R}^n \to \mathbb{R}^m$ uma função diferenciável num conjunto aberto D. Considere-se satisfeita a Hipótese 6.1. Para cada x_{k+1} e $x_k \in L^0 \subset D$ $(x_{k+1} \neq x_k)$*

$$\|c(x_{k+1}) - c(x_k) - A(x_k)(x_{k+1} - x_k)\| \leq \gamma_A \|x_{k+1} - x_k\|^2$$

em que $x_{k+1} = x_k + d_k$.

Na sequência da análise de convergência realizada no Capítulo 5, relativa à direcção pseudo-Newton modificada, d_{PNM_k}, entretanto denotada por d_k, e introduzindo agora o vector correcção $\widetilde{d}_k$, é possível estabelecer condições para garantir que a direcção $d_k + \widetilde{d}_k$ converge para a direcção Newton mais rapidamente do que d_k tende para zero. Assim, o lema seguinte é, para a sucessão $\left\{x_k + d_k + \widetilde{d}_k\right\}$, equivalente ao Lema 5.8 para a sucessão $\{x_k + d_k\}$.

Lema 6.1 *Seja $c(x) : D \subset \mathbb{R}^n \to \mathbb{R}^m$ uma função diferenciável num conjunto aberto D. Considerem-se satisfeitas as Hipóteses 6.1, 6.2 e 6.3 e as condições do Teorema 5.5. Seja $\{x_k\} \subset L^0$ uma sucessão convergente para x^*, tal que $x_{k+1} = x_k + d_k + \widetilde{d}_k$, com d_k definida em (5.35) e $\widetilde{d}_k$ um vector correcção definido por (6.7). Então*

$$\left\|d_k + \widetilde{d}_k - (d_N)_k\right\| = o\left(\|d_k\|\right)$$

em que $(d_N)_k$ é a direcção Newton calculada na iteração k.

Prova. Como

$$\widetilde{d}_k = Y(x_k + d_k)s_k \quad \text{com} \quad L(x_k + d_k)s_k = -c(x_k + d_k)$$

e de acordo com as Hipóteses 6.2 e 6.3 tem-se

$$\|s_k\| \leq r_2 \|c(x_k + d_k)\| \quad \text{e} \quad \left\|\widetilde{d}_k\right\| \leq r_2 \|c(x_k + d_k)\| \tag{6.9}$$

em virtude de Y ser parte de uma matriz ortogonal. Além disso, da Proposição 6.1 tem-se

$$\|c(x_k + d_k)\| - \|c(x_k) + A(x_k)d_k\| \leq \gamma_A \|d_k\|^2 . \tag{6.10}$$

Considere-se agora

$$\begin{aligned}
\|c(x_k) + A(x_k)d_k\| &\leq \|c(x_k) + A(x^*)d_k\| + \|A(x_k)d_k - A(x^*)d_k\| \\
&\leq \|c(x_k) - c(x^*) + A(x^*)d_k\| + \|A(x_k) - A(x^*)\| \, \|d_k\| \\
&\leq \|c(x_k) - c(x^*) - A(x^*)(x_k - x^*) + \\
&\quad + A(x^*)(x_k + d_k - x^*)\| + \gamma_A \|x_k - x^*\| \, \|d_k\|
\end{aligned}$$

pela Hipótese 6.1. Assim, tem-se, usando novamente a Proposição 6.1,

$$\begin{aligned} \|c(x_k) + A(x_k)d_k\| &\leq \gamma_A \|x_k - x^*\|^2 + \|A(x^*)\|_F \|x_k + d_k - x^*\| + \\ &\quad + \gamma_A \|x_k - x^*\| \|d_k\| \\ &\leq \gamma_A (\|x_k - x^*\| + \|d_k\|) \|x_k - x^*\| + \\ &\quad + \widetilde{\gamma} r_1 \|x_k + d_k - x^*\| \end{aligned} \tag{6.11}$$

por (6.8). Portanto, de (6.11)

$$\|c(x_k) + A(x_k)d_k\| = o(\|x_k - x^*\|) \tag{6.12}$$

pois $(\|x_k - x^*\| + \|d_k\|)$ converge para zero quando $k \to \infty$ (pelos Lemas 5.6 e 5.4) e a sucessão $\{x_k\}$, obtida a partir de d_k, converge superlinearmente para x^* (pelo Teorema 5.5). Usando (6.12) em (6.10) tem-se

$$\|c(x_k + d_k)\| \leq \gamma_A \|d_k\|^2 + o(\|x_k - x^*\|)$$

donde se conclui que

$$\|c(x_k + d_k)\| = o(\|d_k\|) \tag{6.13}$$

por (5.53) (Lema 5.7). Considerando agora a relação em (6.9) e usando (6.13) tem-se o seguinte resultado:

$$\left\|\widetilde{d}_k\right\| = o(\|d_k\|). \tag{6.14}$$

Nestas condições, como

$$\left\|d_k + \widetilde{d}_k - (d_N)_k\right\| \leq \|d_k - (d_N)_k\| + \left\|\widetilde{d}_k\right\|$$

conclui-se que

$$\left\|d_k + \widetilde{d}_k - (d_N)_k\right\| = o(\|d_k\|)$$

pelo Lema 5.8. ■

O teorema que se segue estabelece as condições para que a ordem de convergência do processo iterativo definido em (6.6) e (6.7) seja superlinear.

Teorema 6.1 *Seja $\Phi : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ uma função real de variável real continuamente diferenciável no compacto $L^0 \subset D$. Considerem-se as condições do Lema 6.1 e do Teorema 5.5. Seja $\{x_k\} \subset L^0$ uma sucessão definida por $x_{k+1} = x_k + d_k + \widetilde{d}_k$ com d_k dada por (5.35) e $\widetilde{d}_k$ definida por (6.7) onde $\{\mathcal{F}_k\} \in \mathbb{R}^{n\times n}$ é uma sucessão de matrizes não singulares em que*

$$\lim_{k\to\infty} x_k = x^* \in L^0$$

$(x_k \neq x^*)$ para todo o k, $\nabla\Phi(x^*) = 0$ e $\nabla^2\Phi(x^*)$ é não singular. Então $\{x_k\}$ converge para x^* superlinearmente se e só se

$$\left\| d_k + \widetilde{d}_k - (d_N)_k \right\| = o\left(\|d_k\|\right) \tag{6.15}$$

em que $(d_N)_k$ é a direcção Newton calculada em x_k.

Prova. Para simplificar a notação vai usar-se $\nabla^2\Phi_k$, $\nabla\Phi_k$ e $\nabla\Phi_{k+1}$ em vez de, respectivamente, $\nabla^2\Phi(x_k)$, $\nabla\Phi(x_k)$ e $\nabla\Phi(x_k + d_k + \widetilde{d}_k)$. Suponha-se primeiro, que se verifica a condição (6.15). Como

$$\begin{aligned}
\nabla^2\Phi_k\left(d_k + \widetilde{d}_k - (d_N)_k\right) &= \nabla^2\Phi_k\left(d_k + \widetilde{d}_k\right) + \nabla\Phi_k \\
&= \nabla\Phi_{k+1} - \left(\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k\left(d_k + \widetilde{d}_k\right)\right)
\end{aligned}$$

e tomando normas

$$\begin{aligned}
\left\|\nabla^2\Phi_k\left(d_k + \widetilde{d}_k - (d_N)_k\right)\right\| &\geq \|\nabla\Phi_{k+1}\| - \left\|\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k\left(d_k + \widetilde{d}_k\right)\right\| \\
&\geq \|\nabla\Phi_{k+1}\| - \frac{K}{2}\left(\|d_k\| + \left\|\widetilde{d}_k\right\|\right)^2
\end{aligned} \tag{6.16}$$

pela Proposição 5.4. Além disso, como

$$\left\|\nabla^2\Phi_k\left(d_k + \widetilde{d}_k - (d_N)_k\right)\right\| \leq \left\|\nabla^2\Phi_k\right\| \left\|d_k + \widetilde{d}_k - (d_N)_k\right\|$$

tem-se pela Hipótese 5.9

$$\begin{aligned}
\lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k\left(d_k + \widetilde{d}_k - (d_N)_k\right)\right\|}{\|d_k\|} &\leq \lim_{k\to\infty} \left[\left(\left\|\nabla^2\Phi(x^*)\right\| + K\|x_k - x^*\|\right) \frac{\left\|d_k + \widetilde{d}_k - (d_N)_k\right\|}{\|d_k\|}\right] \\
&= 0
\end{aligned}$$

por hipótese. Assim

$$\begin{aligned} 0 &\geq \lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k\left(d_k+\widetilde{d}_k-(d_N)_k\right)\right\|}{\|d_k\|} \\ &\geq \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} - \frac{K}{2}\lim_{k\to\infty}\frac{\left(\|d_k\|+\varsigma_k\|d_k\|\right)^2}{\|d_k\|} \\ &\geq \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} - \frac{K}{2}\lim_{k\to\infty}\left((1+\varsigma_k)^2\|d_k\|\right) \end{aligned}$$

por (6.14) ($\lim_{k\to\infty}\varsigma_k = 0$) e

$$0 \geq \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} \tag{6.17}$$

pelo Lema 5.6. Como por hipótese $\nabla\Phi(x^*) = 0$ e de (5.29)

$$\|\nabla\Phi_{k+1}\| \geq \frac{1}{\gamma_{\nabla\Phi}}\|x_{k+1}-x^*\|$$

e

$$\begin{aligned} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} &\geq \frac{\frac{1}{\gamma_{\nabla\Phi}}\|x_{k+1}-x^*\|}{\|x_k+d_k-x^*\|+\|x_k-x^*\|} \\ &= \frac{\rho_{1k}}{\gamma_{\nabla\Phi}(\rho_{2k}+1)} \end{aligned}$$

com $\rho_{1k} = \dfrac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|}$ e $\rho_{2k} = \dfrac{\|x_k+d_k-x^*\|}{\|x_k-x^*\|}$. Sabe-se do Teorema 5.5 que $\lim_{k\to\infty}\rho_{2k} = 0$.

De (6.17) conclui-se que

$$\lim_{k\to\infty}\frac{\rho_{1k}}{\gamma_{\nabla\Phi}(\rho_{2k}+1)} = 0$$

e $\{\rho_{1k}\}$ converge para zero, o que prova a convergência q-superlinear da sucessão $\{x_k\}$ definida por $x_{k+1} = x_k + d_k + \widetilde{d}_k$. Assuma-se agora que $\{x_k\}$ converge q-superlinearmente para x^*. Como

$$\frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} = \frac{\|\nabla\Phi_{k+1}-\nabla\Phi(x^*)\|\,\|x_k-x^*\|}{\|x_k-x^*\|\,\|d_k\|},$$

por (5.29) tem-se

$$\begin{aligned} \lim_{k\to\infty} \frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} &\leq \lim_{k\to\infty} \left[\frac{\gamma_{\nabla\Phi} \|x_{k+1} - x^*\|}{\|x_k - x^*\|} \frac{\|x_k - x^*\|}{\|d_k\|} \right] \\ &= 0 \end{aligned} \tag{6.18}$$

por hipótese e pelo Lema 5.9. Como

$$-\nabla\Phi_{k+1} = -(\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)) - \nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)$$

tomando normas

$$\begin{aligned} \|\nabla\Phi_{k+1}\| &\geq -\left\|\nabla\Phi_{k+1} - \nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)\right\| + \left\|-\nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)\right\| \\ &\geq \left\|-\nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)\right\| - \frac{K}{2}\left\|d_k + \widetilde{d}_k\right\|^2 \end{aligned} \tag{6.19}$$

pela Proposição 5.4. Além disso

$$\begin{aligned} \left\|-\nabla\Phi_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)\right\| &= \left\|\nabla^2\Phi_k(d_N)_k - \nabla^2\Phi_k(d_k + \widetilde{d}_k)\right\| \\ &= \left\|\nabla^2\Phi_k((d_N)_k - (d_k + \widetilde{d}_k))\right\|. \end{aligned}$$

De (6.19) obtém-se

$$\frac{\|\nabla\Phi_{k+1}\|}{\|d_k\|} \geq \frac{\left\|\nabla^2\Phi_k((d_k + \widetilde{d}_k) - (d_N)_k)\right\|}{\|d_k\|} - \frac{K}{2}\frac{\left\|d_k + \widetilde{d}_k\right\|^2}{\|d_k\|},$$

donde

$$\lim_{k\to\infty} \frac{\left\|\nabla^2\Phi_k((d_k + \widetilde{d}_k) - (d_N)_k)\right\|}{\|d_k\|} = 0$$

por (6.18) e por $\lim_{k\to\infty} \frac{K}{2}\frac{\left\|d_k + \widetilde{d}_k\right\|^2}{\|d_k\|} = 0$. A condição (6.15) surge naturalmente pela equivalência das normas num espaço vectorial de dimensão finita, pois existe uma constante $0 < \gamma < \infty$ tal que

$$\|A\| \leq \gamma \|A\|_M$$

para qualquer matriz de ordem $n \times n$ ou vector de $\mathbb{R}^n$. De facto, considerando $M = \nabla^2 \Phi_k$ não singular (em L^0)

$$\left\| (d_k + \widetilde{d}_k) - (d_N)_k \right\|_M = \left\| M \left((d_k + \widetilde{d}_k) - (d_N)_k \right) \right\|$$

([72, pp 105]) define uma norma para vectores de $\mathbb{R}^n$. Logo

$$\lim_{k \to \infty} \frac{\left\| (d_k + \widetilde{d}_k) - (d_N)_k \right\|_{\nabla^2 \Phi_k}}{\|d_k\|} = \lim_{k \to \infty} \frac{\left\| \nabla^2 \Phi_k \left((d_k + \widetilde{d}_k) - (d_N)_k \right) \right\|}{\|d_k\|} = 0$$

como se pretendia. ■

O lema seguinte é uma consequência da convergência superlinear da sucessão $\{x_k\}$ definida por $x_{k+1} = x_k + d_k + \widetilde{d}_k$.

Lema 6.2 *Seja $\{x_k\}$ a sucessão definida por $x_{k+1} = x_k + d_k + \widetilde{d}_k$. Se $\{x_k\} \subset L^0$ converge superlinearmente para x^* então* $\lim_{k \to \infty} \dfrac{\left\| d_k + \widetilde{d}_k \right\|}{\|x_k - x^*\|} = 1.$

Prova. A prova segue passos idênticos ao da prova do Lema 8.23 em [28] e enunciado como Lema 5.9 no Capítulo 5. ■

6.5 Síntese e comentários finais

Nos últimos anos têm surgido alguns algoritmos que definem a direcção de procura como a resultante de duas componentes. Destes, alguns estão directamente relacionados com um esquema de globalização do tipo procura unidimensional, que garante uma redução significativa do valor de uma função mérito em determinadas iterações do processo iterativo. Outros são implementados no âmbito de uma estratégia de regiões de confiança. Fez-se apenas um resumo do estado da arte dos primeiros, pois são esses que geram uma procura, que na literatura é conhecida por curvilínea.

Neste trabalho, a procura realizada ao longo de um arco considera uma componente que se designa por direcção pseudo-Newton modificada, descrita e analisada no Capítulo 5, e outra designada por vector correcção. Esta componente força a admissibilidade. Mostrou-se que, de acordo com certas condições, a sucessão de aproximações à solução gerada pela resultante das duas componentes, converge q-superlinearmente para um ponto estacionário da função de penalidade exacta usada neste trabalho.

No Capítulo 8 serão apresentadas algumas estatísticas dos resultados numéricos obtidos que mostram que a procura curvilínea é superior (em termos de eficiência e robustez) à procura realizada ao longo de uma linha.

Capítulo 7

Resolução inexacta de sistemas

Este capítulo surgiu pela necessidade de calcular a direcção pseudo-Newton do sistema (5.13), de uma maneira mais eficiente e rápida, especialmente no que diz respeito aos requisitos de recursos informáticos, como memória e tempo de CPU. Um procedimento possível para atingir o objectivo enunciado tem em conta a resolução inexacta do sistema. Faz-se uma descrição sucinta do método dos gradientes conjugados, referindo alguns tópicos sobre a sua convergência. São feitas algumas considerações genéricas sobre as versões MINRES e SYMMLQ para matrizes simétricas e indefinidas e GMRES para matrizes não simétricas. Por fim apresentam-se as ideias básicas do método de Newton truncado e suas características.

7.1 Motivação

Este capítulo vai incidir sobre a resolução de um sistema de equações lineares $Ax = b$, em que A é uma matriz de dimensão $n \times n$ e b um vector de n componentes. Um dos métodos mais conhecidos é o método directo de eliminação de Gauss que requer no máximo um espaço de memória de n^2 para armazenamento da matriz A e envolve $\frac{2n^3}{3}$ operações aritméticas. Sabe-se que um método directo calcula a solução do sistema ao fim de um número fixo de operações aritméticas que pode ser calculado *a priori* e, se a aritmética for exacta, a solução encontrada é também exacta. Para além do método de eliminação de Gauss, existem outras técnicas de factorização da matriz dos coeficientes que também podem ser muito dispendiosas em termos computacionais se a matriz for de grande dimensão. Certas matrizes que surgem nos problemas práticos podem ter algumas propriedades especiais, por exemplo, podem ser esparsas, que não são exploradas pelo método de eliminação de Gauss. Torna-se premente desenvolver métodos de resolução de

sistemas lineares que tirem partido de certas propriedades que a matriz dos coeficientes apresenta.

Os métodos iterativos, ao contrário dos métodos directos, não têm custos fixos uma vez que a solução é obtida de uma sucessão de aproximações à solução, terminando o processo quando se considerar a medida do erro suficientemente pequena relativamente ao problema a resolver. A solução exacta só é encontrada após uma sucessão infinita de aproximações pelo que, o valor obtido é apenas uma aproximação mais ou menos precisa, consoante o erro admitido. Desta forma, a resolução do sistema é inexacta, uma vez que a sucessão infinita de aproximações é interrompida num termo de determinada ordem. Estes métodos são muito usados para resolver sistemas de grandes dimensões e apresentam algumas vantagens relativamente aos métodos directos. Como vantagem salienta-se o facto da matriz não ser factorizada e os requisitos de armazenamento de memória serem muito menores.

Os métodos iterativos dividem-se em estacionários e não estacionários. Os métodos estacionários podem ser postos na forma

$$x_{k+1} = Bx_k + c$$

em que nem a matriz B nem o vector c dependem da iteração k. Exemplos destes métodos são os métodos de Jacobi, Gauss-Seidel, Sobrerelaxação Sucessiva (SOR - *Successive Overrelaxation*) e Sobrerelaxação Sucessiva Simétrica (SSOR - *Symmetric Successive Overrelaxation*). Os livros [21] e [102] facultam informação detalhada sobre este assunto. Nos métodos não estacionários os cálculos computacionais envolvem informação que muda em cada iteração. Exemplos de métodos não estacionários são os métodos baseados em subespaços de Krylov ([1] e [60]) e nos algoritmos ABS ([99]). O método dos gradientes conjugados que vai ser descrito na secção seguinte é um exemplo dos primeiros.

7.2 Método dos gradientes conjugados

Este método, muito conhecido, destina-se à resolução de sistemas lineares e pode ser estendido para resolver problemas de optimização não lineares. Funciona como uma alternativa ao método de eliminação de Gauss, sendo recomendado para sistemas de grandes dimensões. Na literatura, na resolução de sistemas lineares o método chama-se gradientes conjugados lineares.

O método dos gradientes conjugados foi proposto separadamente por Hestenes e por Stiefel nos anos 50 como um método iterativo para a resolução do sistema $Ax = b$ em que A é uma matriz simétrica e definida positiva. Uma revisão histórica, já antiga, que cobre o período de 1948 a 1976, sobre a evolução dos métodos dos gradientes conjugados

e de Lanczos, foi publicada por Golub e O'Leary em 1989 [56] e contém um total de 291 referências. O método é visto como uma técnica para resolver o seguinte problema quadrático de optimização equivalente

$$\begin{array}{ll} min & f(x) \equiv \frac{1}{2} x^T A x - b^T x. \\ x \in \mathbb{R}^n & \end{array}$$

Para o cálculo do ponto estacionário de $f(x)$, tem-se

$$\nabla f(x) = Ax - b = 0,$$

que é um minimizante se a matriz $\nabla^2 f(x) \equiv A$ for definida positiva.

A designação do método vem do facto de gerar uma sequência de vectores $\{p_i\}$ que são conjugados relativamente à matriz dos coeficientes A, ou seja, que verificam

$$p_i^T A p_j = 0 \quad \text{se } i \neq j.$$

Uma vez que a aritmética não é exacta, durante o processo iterativo a propriedade da ortogonalidade nem sempre é verificada pelo que se aconselha a utilização de um processo de reortogonalização como o proposto em [50, pp 108].

Algoritmo 7.1 (Gradientes Conjugados)

Dados x_0;
{calcular $r_0 = b - Ax_0$*}*;
Para $i = 1, 2, ...$
  $z_{i-1} = r_{i-1}$; $\rho_{i-1} = r_{i-1}^T z_{i-1}$
  Se $(i = 1)$ ***Então***
    $p_1 = z_0$
  Senão
    $\beta_{i-1} = \frac{\rho_{i-1}}{\rho_{i-2}}$; $p_i = z_{i-1} + \beta_{i-1} p_{i-1}$
  FimSe
  $q_i = Ap_i$; $\alpha_i = \frac{\rho_{i-1}}{p_i^T q_i}$
  $x_i = x_{i-1} + \alpha_i p_i$
  $r_i = r_{i-1} - \alpha_i q_i$
  {Verificar convergência}
FimPara

O desempenho do método está relacionado com o espectro da matriz A. O método funciona gerando sucessivas aproximações à solução, os resíduos correspondentes e direcções de procura usadas para actualizar as aproximações e os resíduos. Apesar do comprimento destas sequências poder ser grande, apenas é necessário manter em memória um pequeno número de vectores. Em cada iteração, para actualizar os escalares definidos de forma a que as sequências satisfaçam certas condições de ortogonalidade, são efectuados dois produtos internos.

O Algoritmo 7.1 corresponde à utilização do método dos gradientes conjugados na resolução do sistema $Ax = b$.

7.2.1 Convergência

O método dos gradientes conjugados, e considerando aritmética exacta, termina ao fim de n iterações uma vez que as direcções $p_i (i = 1, \ldots, n)$ formam um conjunto de vectores linearmente independentes num espaço a n dimensões. Se a distribuição dos valores próprios de A for favorável o método precisará de menos do que n iterações para encontrar a solução. Como a maioria dos métodos iterativos não converge num número finito de iterações, parece estranho considerar o método dos gradientes conjugados como um método iterativo. Quando o número de variáveis é elevado, dispende-se muito tempo para realizar esse número de iterações e, além disso, como a aritmética não é exacta, a propriedade da terminação quadrática pode não ser satisfeita e o número de iterações ser muito maior que o número de variáveis. Este é o argumento usado para considerar o método dos gradientes conjugados um método iterativo.

Relativamente à convergência, o Teorema 5.5 de [87, pp 115] estabelece que se os valores próprios de A forem vp_i, $i = 1, \ldots, n$, com $vp_1 \leq vp_2 \leq \ldots \leq vp_n$ então a sucessão $\{x_k\}$ de aproximações à solução obedece a

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{vp_{n-k} - vp_1}{vp_{n-k} + vp_1} \right) \|x_0 - x^*\|_A^2 ,$$

sendo o erro medido segundo a norma definida por $\|y\|_A^2 \equiv \frac{1}{2} y^T A y$.

Este teorema pode ser usado para prever o comportamento deste método em problemas específicos. Uma outra expressão de convergência, baseia-se no número de condição de A definido por $K(A) = \|A\|_2 \|A^{-1}\|_2 = \dfrac{vp_n}{vp_1}$. Pode mostrar-se que

$$\|x_k - x^*\|_A \leq \left(\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1} \right)^{2k} \|x_0 - x^*\|_A .$$

Embora esta expressão seja um limite superior do erro, pode ser muito útil quando a única informação que se possui de A é a estimativa de vp_1 e vp_n .

Segundo [85] o método dos gradientes conjugados converge linearmente de acordo com a expressão

$$\frac{\|x_{k+1} - x^*\|_A}{\|x_k - x^*\|_A} \leq \frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1}.$$

Neste método, diminuir o número de condição da matriz implica melhorar a razão de convergência. Esta é a razão que leva os investigadores a introduzirem matrizes de precondição nos algoritmos dos gradientes conjugados. Este assunto será abordado em 7.4. Sobre a convergência também aconselhamos o trabalho de Nocedal [86].

7.2.2 Variantes do método de Lanczos

O método dos gradientes conjugados é eficaz na resolução de sistemas lineares com matriz dos coeficientes simétrica e definida positiva. Em cada iteração são calculados dois produtos internos para actualizar os dois escalares, α e β (veja-se o Algoritmo 7.1), que servem para gerar uma sequência de vectores que satisfazem certas condições de ortogonalidade. Nos sistemas simétricos e definidos positivos a distância entre a aproximação e a solução é mínima, de acordo com alguma norma. Isto é, o iterando x_k é um elemento do espaço definido por $x_0 + span\left\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\right\}$ [1] de tal forma que $(x_k - x^*)^T A(x_k - x^*)$ é mínimo, com x^* a solução exacta de $Ax = b$. Em geral, só é possível garantir a existência deste mínimo quando A é simétrica e definida positiva.

O método dos gradientes conjugados pode ser visto como uma variante especial do método de Lanczos usado no cálculo dos valores e vectores próprios de uma matriz. É fácil deduzir as equações do método dos gradientes conjugados a partir do método de Lanczos (veja-se [1]). Este método gera uma sequência de vectores relacionados com a factorização de uma matriz tridiagonal semelhante à matriz A e que é gerada pelo método de Lanczos.

Quando o método dos gradientes conjugados falha devido ao facto de A ser indefinida, a matriz tridiagonal resultante do método de Lanczos pode aparecer quase singular ou mesmo singular. Quando ela é quase singular a factorização Cholesky (usada pelo método dos gradientes conjugados) é numericamente instável. No caso de ser singular, a aproximação x_k não é definida. Além disso, para matrizes A indefinidas a propriedade relacionada com a minimização não é bem definida.

---

[1] $span\left\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\right\}$ é o $k-$ésimo subespaço de Krylov em $\mathbb{R}^n$ gerado por $r_0 \in \mathbb{R}^n$ e por $A \in \mathbb{R}^{n \times n}$ [1].

Assim, para sistemas lineares com matrizes simétricas e indefinidas, outras variantes do método de Lanczos devem ser usadas. As mais conhecidas geram os métodos de MINRES e SYMMLQ. Estes métodos ultrapassam os inconvenientes dos gradientes conjugados pois usam uma factorização ortogonal do tipo LQ da matriz tridiagonal, que é bem definida mesmo para matrizes quase singulares. Segue-se uma descrição sucinta dos objectivos de cada uma delas. Dado o sistema

$$r = b - Ax, \quad Ar = 0 \tag{7.1}$$

pretende-se calcular aproximações à solução, x, com a forma $x_k = V_k y_k$, para $k \geq 0$, que sejam pontos estacionários da função

$$f_k(y) = (AV_k y - b)^T B (AV_k y - b) \tag{7.2}$$

em que B é uma matriz simétrica, $V_k = [v_1 v_2 \dots v_k]$ onde os v_i $(i = 1, \dots, k)$ formam um conjunto de vectores linearmente independentes. Se B é definida positiva, $f_k(y)$ define a norma do resíduo. Escolhendo $B = A^{-1}$,

$$V_k^T A V_k y_k = V_k^T b, \quad x_k = V_k y_k$$

é ponto estacionário de (7.2) e obtém-se o método SYMMLQ [89]. Por sua vez, tomando $B = I$, minimiza-se $\|r_k\|$ resolvendo

$$V_k^T A^2 V_k y_k = V_k^T A b, \quad x_k = V_k y_k \tag{7.3}$$

e os métodos resultantes chamam-se métodos de resíduos mínimos, de que o MINRES é um exemplo [60]. A experiência tem mostrado [89] que este último não dá resultados tão precisos como o SYMMLQ quando o problema é mal condicionado.

Como o único teste que pode ser usado para terminar as iterações deve ser baseado no tamanho dos resíduos, o método que os minimiza leva, em geral, menos iterações do que os outros. Sugere-se a consulta do trabalho de Paige e Saunders [89] para mais pormenores.

O método dos resíduos mínimos generalizado GMRES é uma extensão do MINRES para sistemas não simétricos. Tal como este, o GMRES gera uma sequência de vectores ortogonais, mas neste caso, ela é baseada no conjunto inteiro dos vectores, que deve ser conservado, uma vez que não existe simetria. Por esta razão existem versões que implementam um recomeço do processo, para torná-lo menos custoso em termos de memória necessária. A primeira versão que surgiu é de Saad e Schultz ([95]) e a mais popular é baseada no processo de ortogonalização de Gram-Schmidt modificado e num processo de recomeço para controlo dos recursos (veja-se em [1]).

7.3 O método de Newton truncado

Além dos aspectos negativos do método de Newton já referidos no Capítulo 2, entretanto ultrapassados pelas modificações introduzidas ao longo deste trabalho, existe um, que não foi até este momento analisado. Está relacionado com a necessidade de resolver em cada iteração, um sistema linear de dimensão n, sendo n o número de variáveis do problema de optimização. Esta questão também se coloca na versão pseudo-modificada do método de Newton, abordada no Capítulo 5.

Uma das maiores vantagens do método de Newton está relacionada com as propriedades locais, designadamente a razão de convergência. Este facto coloca a dúvida sobre a necessidade de resolver o sistema Newton com grande exactidão quando o processo iterativo ainda está longe do ponto estacionário. Além disso, quando o problema é de grande dimensão, por razões já conhecidas, é-se forçado a resolver o sistema por um método iterativo.

No que concerne à exactidão com que o sistema deve ser resolvido, parece natural pensar-se que deve estar relacionada com a proximidade do iterando relativamente a um ponto estacionário. Uma medida que traduz esta ideia é o resíduo relativo, definido por $\frac{\|r_k\|}{\|\nabla\Phi(x_k)\|}$, onde, se d_k é a direcção calculada, então r_k é dado por

$$r_k = \nabla^2\Phi(x_k)d_k + \nabla\Phi(x_k) \tag{7.4}$$

e $\|.\|$ representa qualquer norma em $\mathbb{R}^n$. A ideia que consiste em calcular uma solução aproximada do sistema Newton, por forma a que $\frac{\|r_k\|}{\|\nabla\Phi(x_k)\|}$ seja considerado adequado, origina o método de Newton inexacto. Para clarificar esta ideia mostra-se uma parte do algoritmo que gera a sucessão $\{x_k\}$ de aproximações à solução x^* (Algoritmo 7.2), dada uma aproximação inicial x_0.

A sucessão não negativa $\{\eta_k\}$ é conhecida na literatura como sucessão de influência, do inglês *forcing term*. Dembo, Eisenstat e Steihaug provam que (veja-se em [19]) o método de Newton inexacto é localmente convergente se a sucessão $\{\eta_k\}$ for uniformemente limitada por um. Os autores estabelecem ainda condições que determinam a razão de convergência.

Em [20] Dembo e Steihaug sugerem a utilização dos gradientes conjugados para o cálculo da direcção do sistema Newton. A sucessão de influência $\{\eta_k\}$ controla a exactidão com que o sistema linear Newton vai ser resolvido. A condição em (7.5) é conhecida por critério de paragem do Newton truncado e a direcção d_k que satisfaz (7.5) chama-se direcção Newton truncada.

Algoritmo 7.2 (Método de Newton inexacto)

Dados η_0, x_0;
$k = 0$;
Enquanto *(Critério de paragem = falso)* ***Fazer***
{calcular uma direcção d_k *que satisfaça:*

$$\nabla^2\Phi(x_k)d_k = -\nabla\Phi(x_k) + r_k \quad onde \quad \frac{\|r_k\|}{\|\nabla\Phi(x_k)\|} \leq \eta_k\} \tag{7.5}$$

{$x_{k+1} = x_k + d_k$*}*
{actualizar η_k*}*
$k = k + 1$
FimEnquanto

O teorema que se enuncia de seguida indica como se deve gerar uma sucessão de aproximações $\{x_k\}$ à solução x^*, a partir da direcção Newton truncada que seja convergente com uma razão de convergência entre um e dois.

Teorema 7.1 *(Teorema 2.3 de [20]). Seja* $x_k \to x^*$ *em que* $\nabla^2\Phi(x^*)$ *é definida positiva e suponha-se que* $\nabla^2\Phi(x)$ *é contínua à Lipschitz em* x^*. *Então*

(i) $x_k \to x^*$ *superlinearmente se e só se* $\dfrac{\|r_k\|}{\|\nabla\Phi(x_k)\|} \to 0$ *quando* $k \to \infty$,

(ii) $x_k \to x^*$ *com ordem* $1 + p$ *se e só se* $\limsup\limits_{k\to\infty} \dfrac{\|r_k\|}{\|\nabla\Phi(x_k)\|^{p+1}} < \infty$.

Escolhendo

$$\eta_k = min\left\{\frac{1}{k}, \|\nabla\Phi(x_k)\|^p\right\} \tag{7.6}$$

para algum $0 < p \leq 1$, obtém-se um método de Newton truncado de ordem $p + 1$ ([20]).

A expressão (7.6) é adaptativa. De facto, quando se está ainda longe da solução, os cálculos para a determinação da direcção truncada, d_k, são poucos; quando $\nabla\Phi(x_k) \approx 0$, $\eta_k \approx 0$ e a direcção d_k aproxima-se da direcção Newton em grandeza e direcção.

Nota: Utilizando no critério de paragem do método de Newton truncado a condição

$$\frac{\|r_k\|}{\|\nabla\Phi(x_k)\|} \leq 10^{-10} \tag{7.7}$$

obtêm-se direcções d_k muito próximas da direcção Newton, d_{N_k}, e a redução verificada no número de operações e no tempo de CPU é menos significativa.

Em [32] Eisenstat e Walker propõem duas escolhas para a sucessão de influência que atenuam a situação de sobreresolução do sistema Newton que pode ocorrer se η_k se tornar muito pequeno. Esta dificuldade pode surgir devido ao facto da aproximação local (modelo linear) de $\nabla\Phi(x)$ se afastar demasiado do passo que aproxima a direcção Newton. Quando surge a sobreresolução, $\|\nabla\Phi(x)\|$ não diminui e não há progresso em relação ao mínimo. Os autores referem que conseguiram os melhores resultados computacionais com a seguinte sucessão

$$\eta_k = \gamma_1 \left(\frac{\|\nabla\Phi(x_k)\|}{\|\nabla\Phi(x_{k-1})\|} \right)^{\gamma_2}, \quad k = 1, 2, \ldots \tag{7.8}$$

onde γ_1, γ_2 e η_0 são dados e pertencem aos seguintes intervalos: $\gamma_1 \in [0, 1], \gamma_2 \in [1, 2]$ e $\eta_0 \in [0, 1)$. O trabalho inclui também uma análise de convergência local.

De acordo com o que foi dito em 7.2, a implementação do método dos gradientes conjugados para a resolução do sistema Newton

$$\nabla^2\Phi(x_k)d_k = -\nabla\Phi(x_k)$$

calcula a solução do seguinte problema quadrático

$$\begin{array}{ll} min & \frac{1}{2}d^T\nabla^2\Phi(x_k)d + \nabla\Phi(x_k)^T d, \\ d \in \mathbb{R}^n & \end{array}$$

que será minimizante global se a matriz Hessiana for definida positiva. Assim, só é possível garantir sucesso na implementação da direcção Newton truncada na solução do problema $(P_{s/rest})$ se a matriz Hessiana, $\nabla^2\Phi(x)$, for definida positiva, ao longo de todo o processo iterativo. Esta condição só é garantida numa pequena vizinhança de x^*. Para ser possível implementar com sucesso o método de Newton truncado numa região mais alargada, o algoritmo deve conter um esquema de segurança que identifique situações de conflito. Dembo e Steihaug em [20] propuseram um algoritmo que identifica direcções p, de curvatura negativa, $p^T\nabla^2\Phi(x_k)p < 0$, e outras que verificam $p^T\nabla^2\Phi(x_k)p \approx 0$. Nestas situações o algoritmo termina e a última direcção calculada é usada como aproximação à solução do sistema Newton. Embora esta direcção seja provavelmente muito diferente da direcção Newton, a função diminui ao longo dela. Na primeira iteração, esta estratégia utiliza a direcção de descida máxima $d = -\nabla\Phi$. O algoritmo correspondente é, já alterado para o sistema pseudo-Newton, o seguinte:

Algoritmo 7.3 (Método pseudo-Newton truncado)

Dados x_0, ε; $k = 0$;
{calcular $D\Phi_k$, $D^2\Phi_k$*}*
{testar convergência exterior}
$\{d_0 = 0,\ r_0 = -D\Phi_0,\ p_0 = r_0,\ \delta_0 = r_0^T r_0,\ i = 0\}$
SAIR=Falso
Enquanto *(Não SAIR)* ***Fazer***
$\{q_i = D^2\Phi_k p_i\}$
Se $(p_i^T q_i \leq \varepsilon\delta_i$ ***Então*** $\{$*SAIR=Verdadeiro:* $d = \begin{cases} p_0 & se\ i = 0 \\ d_i & se\ i \neq 0 \end{cases}\}$
Senão
$\{\alpha_i = \dfrac{r_i^T r_i}{p_i^T q_i},\ d_{i+1} = d_i + \alpha_i p_i,\ r_{i+1} = r_i - \alpha_i q_i\}$
Se $(\dfrac{\|r_{i+1}\|}{\|D\Phi_k\|} \leq \eta_k)$ ***Então*** $\{$*SAIR=Verdadeiro:* $d = d_{i+1}\}$
Senão
$\{\beta_i = \dfrac{r_{i+1}^T r_{i+1}}{r_i^T r_i},\ p_{i+1} = r_{i+1} + \beta_i p_i\}$
$\{\delta_{i+1} = r_{i+1}^T r_{i+1} + \beta_i^2 \delta_i,\ i = i + 1\}$
FimSe
FimSe
FimEnquanto

Uma versão mais recente, de 1999, foi publicada por Gould, Lucidi, Roma e Toint [58] e combina o método dos gradientes conjugados com o método de Lanczos e matrizes de precondição na resolução de um subproblema de regiões de confiança de grande dimensão. A integração de matrizes de precondição no método dos gradientes conjugados funciona da mesma maneira que a norma pesada, pela matriz de precondição, usada na definição da região de confiança. A geometria da região é modificada de tal forma que a Hessiana no espaço escalonado é melhor condicionada. Como as matrizes estão relacionadas, é natural considerar-se a mesma matriz. Seja M essa matriz. Como o método dos gradientes conjugados com precondição e o método de Lanczos podem gerar duas bases diferentes para o mesmo espaço de Krylov, o primeiro gerando uma base conjugada em relação a $\nabla^2\Phi$, e o último uma base M-ortonormal, é possível calcular os vectores da base de Lanczos ([60]) a partir dos iterandos do método dos gradientes conjugados. Este relacionamento

foi explorado em [58] originando um algoritmo (Algoritmo 5.1 em [58]) do método de regiões de confiança generalizado de Lanczos, que incorpora passos do método de Lanczos no método dos gradientes conjugados, no cálculo das aproximações à solução, sempre que $p^T\nabla^2\Phi(x_k)p \leq 0$ e a aproximação está dentro da região de confiança, ou se a aproximação está na fronteira ou fora da região de confiança.

Outro algoritmo, com objectivo idêntico ao anterior, que considera uma versão do método dos gradientes conjugados truncado para resolver o subproblema de regiões de confiança que surge da minimização de um problema não linear sem restrições, surgiu em 2000 ([109]), e considera como base o algoritmo de Dembo e Steihaug [20]. Além da condição $d^T B_k d \leq 0$, com B_k uma aproximação simétrica à matriz $\nabla^2\Phi(x_k)$, Yuan também verifica se a direcção calculada pelo método dos gradientes conjugados gera um ponto fora da região de confiança. Nestas situações, o processo termina com o passo mais longo, que se mantém na região de confiança, segundo a direcção d.

Surgiram também, no âmbito dos gradientes conjugados, versões que constroem direcções de procura, a partir do sistema Newton, baseadas em fórmulas com três termos. A vantagem desta alternativa está no facto de não ser necessário implementar procura unidimensional exacta para obter a propriedade da terminação quadrática, isto é, o mínimo de uma função quadrática é calculado após procuras realizadas ao longo de n direcções mutuamente conjugadas. Um dos primeiros trabalhos sobre gradientes conjugados com três termos surgiu em 1983 [30], requer a armazenagem de dois vectores adicionais e considera a primeira direcção (e a direcção de uma iteração de recomeço) arbitrária. Dixon, Ducksbury e Singh em [31] constataram que se a primeira direcção da sequência de direcções conjugadas não é ao longo do gradiente, pode surgir uma direcção nula mesmo que ainda se esteja longe do mínimo. Para ultrapassar esta limitação os autores propõem uma técnica de recomeço e introduzem uma matriz de precondição definida positiva no método. Das condições enunciadas para o recomeço do algoritmo, destaca-se a que verifica se a nova direcção é suficientemente conjugada em relação à direcção inicial de descida máxima. Dos testes numéricos realizados a versão baseada nesta técnica de recomeço sem matriz de precondição foi a que apresentou resultados mais promissores.

Deng e Li ([22]) analisam a convergência global de uma versão do algoritmo dos gradientes conjugados com três termos, que inclui procura unidimensional baseada nas condições de Wolfe-Armijo e na técnica de recomeço das direcções. Introduzindo restrições nos escalares α_k e β_k na fórmula de três termos

$$d_{k+1} = -\nabla\Phi(x_k) + \beta_k d_k + \alpha_k d_{t(p)}, \quad t(p) \leq k+1 \leq t(p+1) \tag{7.9}$$

em que $t(p)$ denota o índice do ponto de recomeço de ordem p, supondo que $\Phi(x)$ é uma função continuamente diferenciável e a aproximação inicial do processo iterativo, x_0, é tal

que o conjunto de nível é limitado, Deng e Li provam a convergência global do algoritmo [22, pp 274].

Fontecilla nos trabalhos [47] e [48] implementa o método dos gradientes conjugados truncado na resolução de um problema de optimização não linear com restrições de igualdade através de um método dos multiplicadores. A sua abordagem tem como ideias centrais a utilização de uma função de x, nomeadamente como a referida em (5.1), para aproximar o vector dos multiplicadores e a resolução não exacta do sistema, para o cálculo da direcção de procura. Este sistema é baseado no gradiente da função Lagrangeana e numa matriz dos coeficientes que é actualizada usando fórmulas de actualização do tipo Quasi-Newton BFGS ou DFP ([43]). A direcção final do algoritmo (Algoritmo 3.2 [48, pp 158]) tem duas componentes. Uma é a direcção de procura Quasi-Newton, d_{QN_k}, e a outra é obtida de

$$v_k = -\nabla c_k (\nabla c_k^T \nabla c_k)^{-1} c_k. \tag{7.10}$$

Assim,

$$d_k = P_k d_{QN_k} + v_k$$

com

$$P_k = I - \nabla c_k (\nabla c_k^T \nabla c_k)^{-1} \nabla c_k^T$$

em que P_k é a projecção ortogonal em $N[\nabla c_k^T]$, o espaço nulo de ∇c_k^T.

Fontecilla [48] considera também a resolução inexacta dos sistemas de dimensão m que resultam do cálculo da função dos multiplicadores (veja-se (5.1)) e do vector v_k em (7.10).

7.4 Precondição

A razão de convergência dos métodos iterativos depende das propriedades espectrais da matriz dos coeficientes, ou seja, da distribuição dos seus valores próprios. A ideia subjacente à técnica de precondição consiste numa transformação do sistema de forma a originar propriedades espectrais mais favoráveis (melhor distribuição dos valores próprios), ou seja, torná-lo mais fácil de resolver. Em vez de se resolver o sistema $Ax = b$ resolve-se o problema equivalente $M^{-1} Ax = M^{-1}b$, em que M é a matriz definida positiva de precondição. A convergência do algoritmo com precondição depende da matriz $M^{-1} A$ e em aritmética exacta converge num número de iterações igual ao número dos seus valores próprios distintos. A razão de convergência depende do número de condição de $M^{-1} A$.

Existe também a precondição à esquerda e à direita em que a matriz de precondição $M = M_1 M_2$ dá origem ao sistema equivalente $M_1^{-1} A M_2^{-1} (M_2 x) = M_1^{-1} b$. Aconselha-se a consulta de [1, pp 40].

Apesar da introdução da precondição originar custos adicionais, existe um compromisso, que varia de problema para problema, entre o custo de construir e aplicar a matriz de precondição (eficiência, esforço computacional e espaço de armazenamento) e o ganho na velocidade de convergência do algoritmo.

Pode-se aplicar a técnica de precondição a alguns métodos iterativos como por exemplo o de Jacobi, o de SSOR e o dos gradientes conjugados. A escolha do tipo de matrizes de precondição não é imediata, depende muito da aplicação e não há nenhuma estratégia genérica mais adequada. Aconselha-se o Capítulo 10 de [60]. Muitas das matrizes de precondição com maior sucesso surgiram no âmbito de classes de problemas especiais em que a origem do problema sugere um tipo particular de precondição, como é caso da discretização de equações diferenciais parciais. Normalmente a matriz de precondição é escolhida de forma a reduzir o número de condição da matriz do sistema e/ou diminuir o seu número de valores próprios distintos. Outra ideia baseia-se na escolha de uma matriz de precondição de forma a que os novos valores próprios fiquem aglomerados, garantindo-se que o número de iterações necessárias para encontrar uma boa aproximação à solução não é muito maior do que o número de aglomerações.

Na implementação de técnicas de precondição em máquinas paralelas há um factor suplementar a ter em conta que é o compromisso entre a eficácia da precondição e a sua eficiência paralela. Muitas das técnicas actuais de precondição têm ainda uma componente sequencial enorme. Em 1996 realizou-se na região de Cooper Mountain nos Estados Unidos da América uma conferência sobre Iterative Methods onde foram apresentados trabalhos sobre as mais variadas versões do método de Lanczos (como as descritas em 7.2.2 e outras), sobre as técnicas de precondição de sistemas e a utilização de computação paralela nos algoritmos. Esta conferência tem-se repetido de dois em dois anos.

Generalizar a aplicação de matrizes específicas de precondição a um conjunto alargado de situações constitui uma área de investigação na qual se deve continuar a investir esforços.

7.5 Comentários e implementação efectuada

A utilização do método dos gradientes conjugados truncado na resolução inexacta do sistema Newton, ou duma aproximação, quer no âmbito de algoritmos de programação quadrática sequencial quer relacionada com as técnicas de penalidade tem originado reduções significativas nos recursos informáticos necessários à sua implementação.

Se não for possível garantir que a matriz dos coeficientes (ou uma sua aproximação) do sistema Newton seja definida positiva, torna-se vantajoso incorporar no método dos gradientes conjugados, um esquema de segurança que assinale as possíveis falhas do mesmo.

A variante do método dos gradientes conjugados truncado que identifica e usa as direcções de curvatura negativa (veja-se o Algoritmo 7.3), foi implementada neste trabalho quer na versão baseada na procura linear quer na procura curvilínea. Na versão linear do método pseudo-Newton modificado, descrita no Capítulo 5 e apresentada no Algoritmo 5.2, apenas na resolução do sistema pseudo-Newton foi implementada a variante truncada. Na versão curvilínea, descrita no Capítulo 6 e apresentada no Algoritmo 6.1, além do sistema pseudo-Newton, também os sistemas que determinam a função dos multiplicadores (em (5.1)), sua pseudo-derivada (em (5.10)) e o cálculo do vector correcção (em (6.7)) foram resolvidos inexactamente pelo mesmo algoritmo dos gradientes conjugados truncado. Nas versões pseudo-Newton modificado, linear truncada e curvilínea truncada, e relativamente à escolha da sucessão de influência, foram realizados testes com as duas condições mais conhecidas, em (7.6) e (7.7). O próximo Capítulo sobre as experiências computacionais efectuadas ao longo deste projecto faz um resumo do desempenho obtido pelas diversas variantes.

Capítulo 8

Experiências computacionais

Este capítulo vai incidir sobre o trabalho experimental efectuado. Começa-se por descrever o ambiente de programação em que as várias versões foram implementadas e relativamente aos resultados apresentam-se dois tipos de tabelas. O primeiro tipo serve para comparação entre várias versões e o segundo contém valores acumulados. Faz-se ainda referência a algumas experiências efectuadas para estudar determinados factores que ao longo do trabalho suscitaram alguma curiosidade.

8.1 Ambiente de programação

A linguagem de programação utilizada para implementar todos os algoritmos foi o Fortran90 no sistema operativo Linux RedHeat versão 5.2. O computador tem as seguintes características: Pentium II Pro 266 Mhz com 65 Mb de memória central.

Os problemas testados, de pequena e média dimensão, fazem parte da base de dados de problemas do CUTE ([4]). A ligação ao CUTE das várias versões implementadas, obrigou à elaboração de *interfaces*, por um lado para poder usar os problemas e por outro, para utilizar as ferramentas que este disponibiliza, como por exemplo, o cálculo automático do gradiente, da matriz Hessiana e/ou das restrições. O código desenvolvido, para todas as versões implementadas, ainda não foi sujeito a procedimentos de optimização.

8.2 Análise de resultados

Esta secção apresenta uma análise dos resultados computacionais obtidos na execução dos vários algoritmos descritos ao longo deste trabalho, para os problemas teste seleccionados.

Problema	n/m	Tipo	Problema	n/m	Tipo	Problema	n/m	Tipo	Problema	n/m	Tipo
AIRCRFTA	8/5	NOR2	DRUGDISE	63/50	LOR2	HS41	4/1	OLR2	LOTSCHD	12/7	QLR2
ALJAZZAF	3/1	QQR2	DTOC1L	58/36	OLR2	HS42	4/2	SQR2	LSNNODOC	5/4	ONR2
ARGTRIG	10/10	NOR2	DTOC1NA	58/36	OQR2	HS46	5/2	OOR2	MARATOS	2/1	QQR2
ARTIF	10/10	NOR2	DTOC1NB	58/36	OQR2	HS47	5/3	OOR2	METHANB8	31/31	NOR2
AVION2	49/15	OLR2	DTOC1NC	58/36	OQR2	HS48	5/2	SLR2	METHANL8	31/31	NOR2
BDVALUE	12/10	NOR2	DTOC1ND	58/36	OQR2	HS49	5/2	OLR2	MINPERM	51/36	LQR2
BOOTH	2/2	NLR2	DTOC3	29/18	QLR2	HS50	5/3	OLR2	MSQRTA	4/4	NQR2
BRATU2D	49/25	NOR2	DTOC4	29/18	QOR2	HS51	5/3	QLR2	MSQRTB	9/9	NQR2
BRATU2DT	49/25	NOR2	EIGMAXC	22/22	LQR2	HS52	5/3	QLR2	ODFITS	10/6	OLR2
BRATU3D	27/1	NOR2	EIGMINC	22/22	LQR2	HS53	5/3	QLR2	OPTCNTRL	32/20	QQR2
BROYDN3D	10/10	NOR2	FCCU	19/8	SLR2	HS54	6/1	OLR2	ORTHREGB	27/6	QQR2
BROYDNBD	10/10	NOR2	GENHS28	10/8	QLR2	HS55	6/6	OLR2	ORTHREGC	25/10	QQR2
BT1	2/1	QQR2	GILBERT	50/1	QQR2	HS56	7/4	OOR2	ORTHREGD	23/10	QOR2
BT10	2/2	LOR2	GRIDNETA	60/36	QNR2	HS6	2/1	QQR2	ORTHREGE	36/20	QOR2
BT11	5/3	OOR2	HAGER1	21/10	SLR2	HS60	3/1	OOR2	PFIT1	3/3	NOR2
BT12	5/3	QQR2	HAGER2	21/10	OLR2	HS61	3/2	QQR2	PFIT2	3/3	NOR2
BT13	5/1	LQR2	HAGER3	21/10	SLR2	HS62	3/1	OLR2	PFIT3	3/3	NOR2
BT2	3/1	QQR2	HAGER4	21/10	OLR2	HS63	3/2	QQR2	PFIT4	3/3	NOR2
BT3	5/3	SLR2	HATFLDG	25/25	NOR2	HS68	4/2	OOR2	PORTFL1	12/1	SLR2
BT4	3/2	QQR2	HEART6	6/6	NOR2	HS69	4/2	OOR2	PORTFL2	12/1	SLR2
BT5	3/2	QQR2	HEART8	8/8	NOR2	HS7	2/1	OOR2	PORTFL3	12/1	SLR2
BT6	5/2	OOR2	HIMMELBE	3/3	NQR2	HS77	5/2	OOR2	PORTFL4	12/1	SLR2
BT7	5/3	OQR2	HONG	4/1	OLR2	HS78	5/3	OOR2	PORTFL6	12/1	SLR2
BT8	5/2	QQR2	HS100LNP	7/2	OOR2	HS79	5/3	OOR2	POWELLBS	2/2	NOR2
BT9	4/2	LOR2	HS104LNP	8/4	OOR2	HS8	2/2	CQR2	POWELLSQ	2/2	NOR2
BYRDSPHR	3/2	LQR2	HS107	9/6	OOR2	HS80	5/3	OOR2	READING5	51/50	OOR2
CBRATU2D	32/8	NOR2	HS111	10/3	OOR2	HS81	5/3	OOR2	RECIPE	3/3	NOR2
CBRATU3D	54/2	NOR2	HS111LNP	10/3	OOR2	HS9	2/1	OLR2	RK23	17/11	LOR2
CHANDHEQ	10/10	NOR2	HS112	10/3	OLR2	HS99	7/2	OOR2	ROBOT	14/2	QOR2
CLUSTER	2/2	NOR2	HS119	16/8	OLR2	HS99EXP	31/21	OOR2	SEMICON1	12/10	NOR2
COOLHANS	9/9	NQR2	HS26	3/1	OOR2	HYDCAR6	29/29	NOR2	SEMICON2	12/10	NOR2
DECONVC	61/1	SQR2	HS27	3/1	SQR2	HYPCIR	2/2	NQR2	TENBARS2	18/8	LOR2
DEGENLPB	20/15	LLR2	HS28	3/1	SLR2	INTEGREQ	52/50	NOR2	TENBARS3	18/8	LOR2
DIXCHLNG	10/5	SOR2	HS39	4/2	LOR2	LCH	300/1	OOR2	TRIGGER	7/6	LOR2
DNIEPER	61/24	QOR2	HS40	4/3	OOR2	LIN	4/2	OLR2	ZANGWIL3	3/3	NLR2

Tabela 8.1: Lista dos 140 problemas do CUTE usados nas experiências

Os nomes destes problemas, bem como a sua dimensão e tipo, de acordo com a classificação definida em Schittkowsky [97], estão descritos na Tabela 8.1.

As tabelas com os resultados exaustivos dos algoritmos testados, que incluem o número de iterações (nit), o número de cálculos da função de penalidade (nf), o número de cálculos do respectivo gradiente (ng), os valores da norma do gradiente da função Lagrangeana ($\|\nabla L\|_2$), da norma das restrições ($\|c\|_2$) e do valor da função objectivo ($f.o.^*$), atingidos no ponto de paragem do processo iterativo, tendo em conta o critério de paragem utilizado, e o tempo de execução dispendido, em milisegundos ($tempo$), encontram-se no Apêndice A.

A dimensão do problema, n/m, refere-se respectivamente ao número de variáveis e ao número de restrições. O tipo do problema é apresentado por uma sequência de quatro caracteres. O primeiro destes caracteres refere-se ao tipo da função objectivo: N - não está definida, C - constante, L - linear, Q - quadrática, S - soma de quadrados e O - geral (nenhum dos casos anteriores). O segundo refere-se ao tipo das restrições e pode tomar os seguintes valores: U - sem restrições, X - variáveis fixas, B - limites simples, L - linear, Q - quadrático, O - geral (nenhum dos casos anteriores). O terceiro indica a suavidade do problema: R - regular, I - irregular. O quarto caracter é um algarismo - 0, 1 ou 2 - e indica o grau mais elevado de derivadas fornecidas analiticamente.

Em todas as implementações houve necessidade de atribuir valores a certas quantidades existentes em fórmulas e condições. Os valores escolhidos basearam-se em experiências anteriores e em critérios racionais. Assim, o critério de paragem nas várias versões implementadas, todas elas baseadas num processo iterativo, consistiu na avaliação da disjunção de dois factores - o número máximo de iterações permitido e a norma dois do gradiente da função Φ, $\|\nabla\Phi\|_2$. Estipulou-se terminar o processo sempre que o número de iterações ultrapassasse o valor cem, no entanto, para alguns problemas este valor foi alargado, sempre que surgiam indícios de que o processo iria convergir. Para limite de $\|\nabla\Phi\|_2$ usou-se 10^{-6}. No cálculo do valor w, em (5.20), quando a direcção pseudo-Newton era quase ortogonal ao pseudo-gradiente, usou-se para ε o valor 0.245 e nas restantes situações utilizou-se o valor 10^{-1}. No Algoritmo 5.2, na condição que avalia se a direcção pseudo-Newton é suficientemente pseudo-descendente, o valor utilizado para $\overline{\eta}_1$ foi 10^{-10}. Nos esquemas de globalização houve também necessidade de atribuir valores a algumas constantes. Assim, o valor utilizado para $\overline{\mu}_1$, na condição de Armijo modificada (Algoritmo 6.1) foi 10^{-7} e o valor de η_2 usado em (3.12) (critério de Curry-Altman-Armijo) foi 10^{-1}. Na procura unidimensional, em todos os algoritmos implementados, o limite do valor de α nas repetidas divisões por dois, δ_1, foi 10^{-4}. No Algoritmo 6.1 o valor utilizado para medir a admissibilidade da aproximação $x_k + d_k$, designado por δ_2, foi 10^{-10}. Relativamente ao parâmetro ξ do Algoritmo 5.1 sobre o procedimento para a determinação do parâmetro de penalidade foi atribuído o valor 10^{-4}.

(90 problemas)	Linear/Armijo	Linear/CAA	Empates
nit	21	29	40
nf	78	12	0
ng	78	12	0
tempo	66	20	4

Tabela 8.2: Tabela de desempenho entre Armijo e CAA

Uma vez que a leitura, nesta fase, das tabelas com todos os resultados obtidos durante a fase experimental tornar-se-ia uma tarefa fastidiosa e a respectiva análise e interpretação pouco claras, optou-se pela apresentação de dois tipos de tabelas, que incluem a síntese dos resultados e estatísticas construídas a partir dos resultados do Apêndice A.

Um tipo de tabela compara dois algoritmos de cada vez e considera o universo de problemas em que pelo menos um dos algoritmos resolve. Esse número aparece no canto superior esquerdo das tabelas. A comparação é feita em termos de determinadas características, que aparecem ao longo das linhas da tabela. As características podem ser: número de iterações, número de cálculos da função e do gradiente e tempo de execução. As duas versões a comparar e os empates verificados entre elas apresentam-se em colunas. No cruzamento duma linha com uma coluna é apresentado o número de problemas que a versão em coluna, relativamente à característica em linha, foi melhor que a outra versão. Relativamente às características apresentadas, melhor significa que originou menores valores de nit, nf, ng, ou *tempo*. A coluna dos empates, relativamente a determinada característica, representa o número de problemas em que as duas versões apresentam o mesmo desempenho.

O outro tipo de tabela compara três ou quatro algoritmos duma vez e contém os valores acumulados das características analisadas para um determinado universo de problemas. Neste caso o universo refere-se aos problemas resolvidos simultaneamente por todos os algoritmos em análise. O cruzamento de uma linha com uma coluna representa o somatório, para esse universo de problemas, dos valores da referida característica (em linha) obtidos pela implementação do algoritmo (em coluna).

O primeiro conjunto de tabelas pretende comparar esquemas de globalização, descritos no Capítulo 3 e utilizados no método pseudo-Newton modificado baseado numa procura linear. São apresentadas, para o mesmo algoritmo de cálculo da direcção pseudo-Newton modificada baseada numa procura linear, três versões para cálculo do tamanho do passo α - Linear/Armijo (baseada na condição (3.5)), Linear/CAA (definida no Algoritmo 3.1) e Linear/NMonótona (definida no Algoritmo 3.2).

Pode concluir-se da Tabela 8.2 (síntese das Tabelas A.1 e A.2 do Apêndice A), que a versão Linear/Armijo, embora quanto ao número de iterações não apresente vantagens

(89 problemas)	Linear/Armijo	Linear/NMonótona	Empates
nit	4	6	79
nf	2	14	73
ng	4	6	79
tempo	2	18	69

Tabela 8.3: Tabela de desempenho entre Armijo e NMonótona

(75 problemas)	Linear/Armijo	Linear/CAA	Linear/NMonótona
nit	948	1343	963
nf	2073	3339	1627
ng	1023	2761	1038
tempo	285008	193093	277669

Tabela 8.4: Valores acumulados das versões lineares

relativamente à versão Linear/CAA, é indiscutivelmente melhor no número de cálculos da função e do gradiente (87% dos problemas) e em tempos de execução (73% dos problemas).

As versões Linear/Armijo e Linear/NMonótona, na Tabela 8.3 (síntese das Tabelas A.1 e A.3 do Apêndice A) apresentam desempenhos semelhantes em 89% de problemas no que diz respeito a número de iterações e de cálculos do gradiente. Nos restantes 11% a versão Linear/NMonótona foi melhor. Em 82%, o desempenho é semelhante em relação ao número de cálculos da função e em 78% os tempos são comparáveis.

Através da análise da Tabela 8.4 dos valores acumulados, pode concluir-se que as versões Linear/Armijo e Linear/NMonótona apresentam valores semelhantes para todas as características com excepção do número de cálculos da função. Estas versões, no que concerne às características nit, nf e ng, apresentam melhor desempenho do que a versão Linear/CAA, evidenciando esta um melhor desempenho relativamente ao tempo. Embora na Tabela 8.2, a versão Linear/CAA apresente melhores tempos em apenas vinte problemas quando comparada com os sessenta e seis da versão Linear/Armijo, a análise dos valores acumulados permite concluir que, a diminuição dos tempos nesses vinte problemas é de tal forma significativa que, no cômputo geral origina tempos acumulados menores (193093) do que os da Linear/Armijo (285008), não obstante esta última apresentar tempos menores para um maior número de problemas.

Um factor importante a considerar é o número de problemas resolvidos apenas por uma versão quando comparada com outra. Assim, a versão Linear/Armijo resolve onze problemas que a Linear/CAA não resolve e esta resolve cinco problemas que a Linear/Armijo não resolve. A versão Linear/NMonótona resolve um problema que a Linear/Armijo não

(101 problemas)	Linear/Armijo	Curvilínea	Empates
nit	19	46	36
nf	31	33	37
ng	32	34	35
tempo	26	33	42

Tabela 8.5: Tabela de desempenho entre Linear e Curvilínea

resolve. Ainda ficam quarenta e quatro problemas que não são resolvidos por nenhuma das versões.

Fizeram-se ainda algumas experiências computacionais com a versão Linear/Armijo implementando uma outra fórmula para o cálculo do ω, que define a direcção modificada, exigindo que a nova direcção seja suficientemente descendente (Definição 2.2) para $\eta_1 = 10^{-10}$. Convém relembrar que a fórmula ω em (5.20) surge da imposição de descendência. O Teorema 5.2 estabelece as condições que fazem com que a direcção pseudo-Newton modificada se torne suficientemente descendente (veja-se em i)). Apenas nos problemas BT8 e HS68 se verificaram melhorias.

As Tabelas 8.5 (síntese das Tabelas A.1 e A.4), 8.7 (síntese das Tabelas A.4 e A.6) e 8.8 contêm os dados referentes à comparação entre as versões Linear, em que apenas se usa a direcção pseudo-Newton modificada, d_{PNM}, e a Curvilínea, que introduz, além disso, o vector correcção $\widetilde{d}$. Na versão Curvilínea implementada, sempre que é necessário introduzir o vector correcção $\widetilde{d}$, na globalização do algoritmo utiliza-se o esquema Preditor-Corrector-Curry-Altman-Armijo (PCCAA) descrito no Algoritmo 6.2.

A versão Curvilínea parece ter um melhor desempenho do que a Linear, sendo especialmente notório no que diz respeito ao número de iterações. Tendo em conta que o universo de problemas é cento e um e que houve trinta e seis empates, relativamente a *nit*, a percentagem de problemas com menor número de iterações para a versão Curvilínea é mais do dobro do que a da versão Linear.

Experimentou-se ainda na versão Curvilínea a condição de Armijo modificada (6.5) para o cálculo do passo de procura α (veja-se a Tabela A.6). Do universo de problemas em que houve convergência para as duas versões, a que usa a condição de Armijo modificada e a que usa o critério PCCAA, consideraram-se apenas os problemas em que os desempenhos foram diferentes (quarenta e sete problemas), ou seja, não se contabilizaram os problemas que apresentavam o mesmo número de iterações e cálculos da função nas duas versões. Elaborou-se a Tabela 8.6 que contém os valores médios para as características *nit* e *nf* .

Desta tabela conclui-se que em termos médios, a versão Curvilínea com o critério PCCAA apresentou resultados substancialmente melhores, pelo que se decidiu considerá-la como referência para as comparações efectuadas a partir daqui.

(47 problemas)	Curvilínea com Armijo	Curvilínea com PCCAA
$nit_{méd}$	21	12
$nf_{méd}$	40	23

Tabela 8.6: Valores médios de nit e nf para a versão Curvilínea com Armijo e com PCCAA

(91 problemas)	Curvilínea	Curvilínea/NMonótona	Empates
nit	21	0	70
nf	23	0	68
ng	22	1	68
tempo	22	1	68

Tabela 8.7: Tabela de desempenho entre Curvilínea e Curvilínea/NMonótona

Como se pode ver na Tabela 8.7, a introdução do esquema de globalização não monótono na versão Curvilínea (versão Curvilínea/NMonótona) não trouxe qualquer vantagem para o desempenho do Algoritmo 6.1.

Observe-se agora a Tabela 8.8 de valores acumulados. Esta tabela realça o desempenho da versão Curvilínea relativamente à Linear. O número de iterações e cálculos da função é reduzido para menos de metade na versão Curvilínea. Ainda relativamente ao número de cálculos do gradiente e ao tempo dispendido a melhoria é substancial.

A versão Linear/Armijo resolve dez problemas que a Curvilínea não resolve. Porém a Curvilínea resolve catorze problemas que a Linear/Armijo não resolve. A versão Curvilínea resolve catorze problemas que a Curvilínea/NMonótona não resolve. Deste conjunto de experiências, ficam cinquenta e dois problemas por resolver.

Os resultados obtidos pelo conjunto de tabelas anteriores, permitiram concluir que a versão Curvilínea apresenta melhores resultados do que qualquer outra implementada neste trabalho. Por este facto, decidiu-se investigar o seu desempenho, comparando-a com os pacotes de *software* NPSOL e MATLAB, já referidos no Capítulo 1. Os resul-

(70 problemas)	Linear/Armijo	Curvilínea	Curvilínea/NMonótona
nit	1337	530	662
nf	2303	1030	1342
ng	1406	858	1123
tempo	382009	245898	248408

Tabela 8.8: Comparação com valores acumulados da Curvilínea

(115 problemas)	Curvilínea	NPSOL	Empates
nit	87	28	0
nf	67	44	4

Tabela 8.9: Tabela de desempenho entre Curvilínea e NPSOL

(56 problemas)	Curvilínea	NPSOL
nit	380	880
nf	741	1232

Tabela 8.10: Valores acumulados

tados exaustivos dos testes efectuados com estes dois pacotes encontram-se no Apêndice A, nas Tabelas A.7 e A.8, respectivamente, e foram obtidos utilizando-se os parâmetros apresentados por defeito. A comparação em termos de tempos de execução não foi efectuada, uma vez que não houve neste trabalho optimização de código para os algoritmos implementados, o que não acontece com estes dois pacotes de *software* comercializável. Não faria sentido, nem sequer seria justo, efectuar uma comparação em termos de tempos, fazendo-se apenas a comparação com o NPSOL em termos de número de iterações e cálculos da função, e em número de cálculos da função com o MATLAB.

A Tabela A.7 (três últimas colunas) do Apêndice A contém os resultados obtidos pela implementação da versão Curvilínea com PCCAA, tendo como critério de paragem o mesmo do NPSOL. Este baseia-se na conjunção das seguintes condições:

$$
\begin{array}{ll}
i) & \|x_{k+1} - x_k\| \leq \sqrt{1.5 \times 10^{-8}}(1 + \|x_{k+1}\|) \\
ii) & \left\|Z^T(x_{k+1})\nabla f(x_{k+1})\right\| \leq \sqrt{1.5 \times 10^{-8}}(1 + \max(1 + |f(x_{k+1})|, \|\nabla f(x_{k+1})\|)) \\
iii) & \max_i |c_i(x_{k+1})| \leq 1.5 \times 10^{-8}
\end{array}
$$

em que a matriz $Z \in \mathbb{R}^{n\times(n-m)}$ é a matriz das $(n-m)$ últimas colunas de Q, sendo $Q \in \mathbb{R}^{n\times n}$ a matriz ortogonal resultante da factorização QR baseada no método de Householder da matriz ∇c (ver Secção 6.2). O resumo dos resultados obtidos é apresentado na Tabela 8.9 (síntese da Tabela A.7) e não considerando os empates, conclui-se que a versão Curvilínea apresenta melhor desempenho em 76% dos problemas para a característica *nit* e em 60% para nf, relativamente ao NPSOL. A versão Curvilínea resolve trinta e cinco problemas que o NPSOL não resolve e o NPSOL resolve dezanove problemas que a versão Curvilínea não resolve. Vinte e quatro problemas não são resolvidos por nenhum dos dois códigos.

Embora haja convergência simultanea dos dois códigos em sessenta e um problemas, foram retirados os problemas BT5, HS111, HS111LNP, HS26 e LCH, ficando um universo

(117 problemas)	Curvilínea	MATLAB	Empates
nf	51	58	8

Tabela 8.11: Tabela de desempenho entre Curvilínea e MATLAB

(68 problemas)	Curvilínea	MATLAB
nf	1267	1326

Tabela 8.12: Valores acumulados

de cinquenta e seis problemas, pelo facto de apresentarem valores excessivos de nit e nf que, se fossem contabilizados na Tabela 8.10 de valores acumulados, adulterariam a tendência verificada da grande maioria dos resultados. Assim, em termos do número de iterações acumuladas, a versão Curvilínea apresenta um valor inferior a metade do número de iterações acumuladas obtidas pelo NPSOL. Relativamente ao número de cálculos da função, esta redução é também significativa.

A Tabela A.8 (duas últimas colunas) do Apêndice A contém os resultados obtidos pela implementação da versão curvilínea com PCCAA, tendo como condições no critério de paragem as mesmas do MATLAB. O critério de paragem utilizado no MATLAB baseia-se no número máximo de iterações permitido ($100n$) ou na conjunção das três condições seguintes: erro em x inferior a 10^{-7}, erro na função objectivo inferior a 10^{-7} e pior violação das restrições inferior a 10^{-9}. A Tabela 8.11 (síntese da Tabela A.8) compara a versão Curvilínea com o MATLAB em termos de número de cálculos da função: em 47% dos problemas, a versão Curvilínea mostrou-se superior. A versão Curvilínea resolve vinte problemas que o MATLAB não resolve e o MATLAB resolve vinte e oito problemas que a versão Curvilínea não resolve. Deste conjunto de experiências ficam ainda vinte e dois problemas por resolver.

Nesta comparação existem sessenta e nove problemas em que há convergência simultanea nos dois códigos mas, foi retirado o problema HATFLDG, pela mesma razão apontada a quando da comparação da versão Curvílinea com o NPSOL, ficando com sessenta e oito problemas. Os valores acumulados estão na Tabela 8.12 e pode constatar-se que os valores acumulados para o cálculo da função são ligeiramente menores na versão Curvilínea. Existem ainda 16 problemas que não são resolvidos por nenhum dos três códigos. Estes valores fundamentam o facto, muito conhecido no domínio da optimização não linear, de que não existe nenhum algoritmo que seja o mais eficiente e robusto.

A Tabela 8.13 (síntese das Tabelas A.1 e A.9) compara o desempenho da versão básica Linear, com a correspondente versão truncada. No Capítulo 7 foi analisada a estratégia relacionada com a resolução não exacta de sistemas lineares, nomeadamente do sistema

(100 problemas)	Linear	Linear/Truncada	Empates
nit	38	19	43
nf	39	20	41
ng	28	19	43
tempo	48	31	21

Tabela 8.13: Tabela de desempenho entre Linear e Linear/Truncada

pseudo-Newton definido em (5.13). A implementação do método dos gradientes conjugados no cálculo de uma aproximação à solução deste sistema origina a versão Linear truncada (Linear/Truncada). Na condição de paragem do algoritmo truncado (veja-se em (7.3)) usou-se para a sucessão de influência η_k, o valor 10^{-10} (veja-se a Tabela A.9). Neste caso a resolução do sistema origina uma aproximação à direcção pseudo-Newton quase exacta. Da Tabela 8.13 pode concluir-se que a versão truncada não traz qualquer vantagem na resolução de problemas de pequena e média dimensão. Também foram realizadas experiências com a fórmula definida em (7.6) (veja-se as três últimas colunas da Tabela A.9). Neste caso, a resolução do sistema é inexacta. A aproximação à solução do sistema pseudo-Newton está tanto mais próxima quanto mais perto o processo iterativo externo estiver do ponto estacionário da função de penalidade.

Experimentou-se ainda, na versão Linear/Truncada a introdução de uma matriz de precondição. Optou-se por uma matriz diagonal com elementos iguais aos da diagonal principal da matriz dos coeficientes do sistema em valor absoluto, mas não se registaram melhorias substanciais que justificassem a elaboração de uma tabela.

Testou-se uma alteração ao critério de paragem do Algoritmo 7.3, introduzindo a fórmula (7.8) proposta por Eisenstat e Walker em [32] para o cálculo dum limite da sucessão de influência η_k. Os valores utilizados para as constantes γ_1 e γ_2 foram respectivamente 1 e $\frac{1+\sqrt{5}}{2}$. Apenas os problemas BT9, HS100LNP e HS39 mostraram melhor desempenho, passando-se o contrário em cinquenta e quatro problemas e nos restantes os desempenhos foram semelhantes.

A versão truncada foi estendida ao Algoritmo 6.1 de procura curvilínea originando a versão Curvilínea truncada (Curvilínea/Truncada). Desta vez, além do sistema pseudo-Newton, também o sistema do cálculo da estimativa de λ (5.1), os sistemas do cálculo da matriz $D\lambda$ (5.10) e o cálculo do vector correcção $\widetilde{d}$ (6.3) foram resolvidos pelo método dos gradientes conjugados truncado. As Tabelas A.10 e A.11 contêm estas versões truncadas respectivamente para $\eta_k = 10^{-10}$ e η_k dado por (7.6).

A observação da Tabela 8.14 (síntese das Tabelas A.4 e A.10) permite concluir que existe um certo equilíbrio entre as duas versões comparadas, ou seja, nenhuma versão

(105 problemas)	Curvilínea	Curvilínea/Truncada	Empates
nit	38	40	27
nf	49	35	21
ng	52	32	21
tempo	42	47	16

Tabela 8.14: Tabela de desempenho entre Curvilínea e Curvilínea/Truncada

(61 problemas)	Linear/Armijo	Linear/Truncada	Curvilínea	Curvilínea/Truncada
nit	933	1129	1103	567
nf	1754	3702	2173	1773
ng	994	1187	2138	1106
tempo	332500	147082	167564	26640

Tabela 8.15: Valores acumulados das versões truncadas

parece ser significativamente melhor do que a outra. A versão Curvilínea apresenta uma vantagem relativamente ao número de cálculos da função e do gradiente, mas a versão Curvilínea truncada leva, no cômputo global, menos iterações e menos tempo.

Considere-se agora a Tabela 8.15 de valores acumulados. Pela simples observação da Tabela 8.15, constata-se de imediato que a versão Curvilínea/Truncada apresenta menor número de iterações e uma redução do tempo de execução de 82 a 92%, relativamente às outras versões. Um facto interessante a salientar é que na Tabela 8.13, embora a versão Linear/Truncada origine, em geral, mais iterações (externas), estas são muito mais rápidas do que as da versão Linear.

Da mesma tabela se infere que a utilização da técnica de truncagem na versão Curvilínea tem influência muito positiva no desempenho dos algoritmos, relativamente às quatro características analisadas. Convém no entanto salientar, que é em relação ao tempo de execução que esta vantagem é mais substancial.

À semelhança das análises efectuadas para os conjuntos de tabelas anteriores, a versão Linear resolve dezassete problemas que a Linear/Truncada não resolve e esta resolve doze problemas que a Linear não resolve. Relativamente à versão Curvilínea, esta resolve dezassete problemas que a Curvilínea/Truncada não resolve e não resolve treze problemas que apenas são resolvidos pela versão Curvilínea/Truncada. Vinte e quatro problemas não foram resolvidos por nenhuma destas duas versões.

Após se ter constatado que a utilização da técnica de truncagem, no algoritmo de procura curvilínea, pode melhorar a eficiência dos algoritmos, procedeu-se à selecção de onze problemas de média dimensão e de oito problemas de pequena dimensão, para se

Problema	Linear	Linear/Truncada	Curvilínea	Curvilínea/Truncada
BRATU2D	10^5 (18/115)	3825 (2/3)	2.7×10^4 (3/22)	128 (1/3)
BRATU2DT	1.9×10^5 (24/163)	6385 (4/5)	5099 (3/4)	21 (2/5)
CBRATU3D	208 (1/2)	486 (4/5)	207 (1/2)	30 (2/5)
DTOC1NA	1.7×10^5 (52/53)	10^5 (15/34)	9.4×10^4 (22/31)	1.2×10^4 (29/103)
DTOC4	10^3 (4/5)	10^3 (4/5)	1.46×10^3 (3/7)	171 (3/7)
EIGMAXC	10^3 (6/8)	2720 (10/19)	1235 (7/8)	237 (4/9)
HATFLDG	4430 (6/20)	4450 (6/20)	3.8×10^5 (813/1627)	2373 (88/176)
INTEGREQ	–	10^4 (2/3)	2.7×10^4 (3/7)	364 (1/3)
OPTCNTRL	4826 (14/15)	2×10^4 (43/74)	6213 (10/20)	8×10^3 (153/743)
READING5	3.5×10^4 (1/10)	10^4 (1/3)	5.9×10^4 (1/17)	2×10^3 (1/3)
SEMICON1	31 (1/2)	44 (2/3)	33 (1/2)	13 (1/3)

Tabela 8.16: Valores de "tempo (nit/nf)" para onze problemas de média dimensão

Problema	Linear/Armijo	Linear/Truncada	Curvilínea	Curvilínea/Truncada
BT10	27 (6/8)	8 (6/8)	9 (4/5)	21 (5/25)
HS28	5 (1/2)	52 (1/2)	8 (1/2)	7 (1/2)
HS55	2499 (149/538)	187 (2/3)	198 (3/22)	25 (6/8)
HS60	19 (35/36)	466 (35/36)	24 (25/42)	104 (32/52)
HS8	7 (4/6)	183 (4/6)	10 (7/8)	15 (3/6)
HS80	102 (113/114)	1165 (113/114)	21 (8/17)	30 (7/15)
MARATOS	11 (17/18)	10 (17/18)	11 (5/9)	23 (6/10)
POWELLBS	29 (21/93)	14 (19/31)	11 (9/10)	33 (7/60)

Tabela 8.17: Valores de "tempo (nit/nf)" para oito problemas de pequena dimensão

analisar a influência da dimensão do problema nestas técnicas. A observação da Tabela 8.16, relativa a problemas de média dimensão, permite concluir que para as versões Curvilíneas, mesmo quando o número de iterações e de cálculos de função é semelhante, os tempos de execução são significativamente menores na versão truncada. Com as versões Lineares isto já não acontece.

No entanto, a Tabela 8.17, para problemas de pequena dimensão, mostra que a observação feita à Tabela 8.16 relativamente às versões Curvilíneas, deixa de fazer sentido para os problemas de pequena dimensão, ou seja, no geral não há uma diminuição visível dos tempos de execução na versão truncada.

A análise destas duas tabelas indicia que, a resolução não exacta dos sistemas lineares,

(100 problemas)	Curvilínea	Curvilínea/Truncada (7.6)	Empates
nit	62	32	6
nf	68	27	5
tempo	86	14	0

Tabela 8.18: Tabela de desempenho entre Curvilínea e Curvilínea/Truncada (7.6)

(51 problemas)	Curvilínea	Curvilínea/Truncada (7.7)	Curvilínea/Truncada (7.6)
nit	301	385	291
nf	598	985	930
tempo	14277	2296	18210

Tabela 8.19: Valores acumulados

por métodos iterativos, na versão Curvilínea, é mais vantajosa para problemas de maiores dimensões, podendo mesmo prejudicar o desempenho do algoritmo no caso de problemas de pequenas dimensões. Esta constatação não traz nada de novo uma vez que, a literatura especializada no domínio da resolução de sistemas, aconselha a utilização de técnicas de truncagem apenas nos problemas de média ou grande dimensão.

A Tabela 8.18 (síntese das Tabelas A.4 e A.11) compara a versão Curvilínea com a Curvilínea/Truncada baseada na condição adaptativa (7.6). A resolução do sistema pseudo-Newton, e dos outros atrás descritos, é feita de tal forma que, quanto mais próximo do ponto estacionário estivermos, mais próximo da direcção Newton está a aproximação calculada pelo método dos gradientes conjugados truncado. A Tabela 8.19 dos valores acumulados mostra que surgem reduções no número de iterações e no número de cálculos da função, quando a condição adaptativa é utilizada, mas, no cômputo geral o processo é mais lento do que a versão não truncada do algoritmo de procura curvilínea, sendo oito vezes mais lento do que a versão que utiliza o valor $\eta_k = 10^{-10}$.

A versão Curvilínea resolve trinta e três problemas que a Curvilínea/Truncada (7.6) não resolve e esta resolve dez problemas que a Curvilínea não resolve. Quarenta problemas continuam a não ser resolvidos por nenhuma das duas versões.

Capítulo 9

Conclusões e investigação futura

No final de qualquer projecto é importante fazer um balanço sobre o trabalho realizado para verificar se os objectivos propostos foram alcançados, identificando as principais dificuldades encontradas e os aspectos positivos e negativos da sua realização. Esta reflexão permite, não só avaliar o esforço dispendido, mas também definir linhas de acção futuras, de forma a complementar o trabalho e estender a investigação em áreas afins.

9.1 Conclusões

Os resultados obtidos neste trabalho podem ser aferidos sob duas perspectivas - o ponto de vista prático, que se traduz na avaliação do desempenho dos algoritmos desenvolvidos na resolução de problemas teste, e a componente teórica que valida esses mesmos algoritmos.

O problema analisado neste projecto, problema de optimização não linear com restrições de igualdade, foi resolvido utilizando-se uma técnica de penalidade exacta. Neste trabalho, o parâmetro de penalidade foi escolhido e ajustado através dum procedimento adaptativo, já utilizado em estudos anteriores, tendo o seu comportamento sido no cômputo geral satisfatório.

No cálculo do ponto estacionário da função de penalidade exacta baseada numa Lagrangeana aumentada, recorreu-se ao método de Newton, tendo-se aproximado a matriz Hessiana e o vector gradiente, respectivamente pela pseudo-Hessiana e pelo pseudo-gradiente, dando origem ao sistema pseudo-Newton, cuja direcção de procura se denominou direcção pseudo-Newton. Este sistema pode ser reescrito de forma a evidenciar a sua correspondência com o sistema Newton em que apenas a matriz dos coeficientes é modificada. A complexidade das fórmulas foi substancialmente reduzida, permitindo diminuir tempos de execução e espaço de memória. Provou-se que este método é, em termos lo-

cais e sob certas condições, pelo menos q-linearmente convergente, podendo nalguns casos específicos apresentar convergência q-quadrática.

As limitações que podem surgir durante a implementação do método de Newton persistiam também no método pseudo-Newton, obrigando à definição duma nova direcção, agora sempre descendente, que foi designada por direcção pseudo-Newton modificada (d_{PNM}). Foram estabelecidas condições que tornam esta direcção suficientemente descendente, do tipo gradiente, limitada em norma, dando origem a uma redução significativa no valor da função de penalidade. O parâmetro que caracteriza a função $w(x)$, na definição da direcção modificada, faz o equilíbrio entre a direcção pseudo-Newton e a direcção de pseudo-descida máxima, observando-se em certos problemas alguma sensibilidade a este parâmetro. Provou-se que $\|d_{PNM} - d_N\| = o(\|d_{PNM}\|)$, em que d_N é a direcção Newton, e é baseado neste resultado que se demonstra a convergência q-superlinear do método pseudo-Newton modificado.

Tendo havido necessidade de introduzir uma técnica de globalização que garantisse convergência foram utilizados esquemas que se baseiam numa filosofia de procura unidimensional. Além de uma modificação introduzida na condição de Armijo, apropriada ao sistema pseudo-Newton, foi desenvolvida outra estratégia de escolha de um comprimento de passo inicial, designada Curry-Altman-Armijo (CAA). Observou-se que para certos problemas o critério de Armijo modificado funcionou melhor enquanto que para outros o CAA foi mais eficiente.

Como o método baseado na função de penalidade é de pontos exteriores à região admissível e no sentido de acelerar a convergência do método, definiu-se uma direcção $\widetilde{d}$, por forma a encaminhar o processo iterativo no sentido da região admissível. O passo final, neste caso, é caracterizado por duas componentes, d_{PNM} e $\widetilde{d}$. A introdução da estratégia de procura unidimensional define uma procura curvilínea ao longo da resultante das duas componentes. A condição de Armijo e um novo critério denominado Preditor-Corrector-Curry-Altman-Armijo foram estendidos a este caso. Provou-se que $\left\|\widetilde{d}\right\| = o(\|d_{PNM}\|)$ e ainda que $\left\|d_{PNM} + \widetilde{d} - d_N\right\| = o(\|d_{PNM}\|)$, garantindo-se consequentemente uma convergência q-superlinear. Com este algoritmo obtiveram-se, em termos gerais, os melhores resultados de todas as experiências computacionais realizadas, pelo que serviram de termo comparativo com outros códigos conhecidos.

Embora os problemas testados não fossem de grande dimensão e não apresentassem nenhuma estrutura específica, houve a preocupação de resolver os sistemas lineares de uma forma eficiente minimizando os custos computacionais. Utilizou-se o método dos gradientes conjugados truncado com duas variantes para a sucessão de influência na paragem do processo iterativo. Os resultados apontam para uma redução nos tempos de execução mesmo quando não se verificam reduções no número de iterações e cálculos da

função. Esperar-se-iam valores mais conclusivos, principalmente em termos de tempo de execução, se a dimensão dos problemas fosse maior.

Uma das motivações deste trabalho era desenvolver um algoritmo que não usasse a filosofia subjacente à programação quadrática sequencial, utilizada nomeadamente nos pacotes MATLAB e NPSOL, e que pudesse ser comparável, em termos de eficiência e robustez, com estes códigos. De facto, recorrendo à técnica de penalidade exacta foram efectuadas comparações, em termos de testes computacionais, entre as várias versões implementadas e o MATLAB e NPSOL. Em termos gerais pode afirmar-se que, o método pseudo-Newton modificado e truncado com procura curvilínea foi, no universo das versões desenvolvidas, a que apresentou melhor desempenho. A comparação com o MATLAB e NPSOL permite afirmar que o código desenvolvido, baseado na técnica de penalidade exacta, é comparável aos códigos que utilizam a programação quadrática sequencial.

9.2 Futuras direcções de investigação

Quando se reflecte sobre o que ainda é possível fazer para melhorar ou completar um projecto de investigação, muitas são as ideias que surgem e as dúvidas que persistem. Este projecto de doutoramento não é excepção. Enumeram-se de seguida, alguns temas relacionados com o trabalho desenvolvido que vão merecer, num futuro próximo, alguma atenção e dedicação.

Numa primeira fase pretende-se explorar a utilização dos métodos iterativos de Krylov na resolução inexacta de sistemas. Estes métodos têm surgido na literatura, sendo o método dos gradientes conjugados o mais utilizado. Pelo facto de não ser possível garantir que a matriz dos coeficientes do sistema pseudo-Newton seja definida positiva, a versão do método dos gradientes conjugados que se baseia na factorização LQ de uma matriz tridiagonal simétrica [89], que surge do processo de Lanczos, irá provavelmente tornar o algoritmo final mais robusto.

Pretende-se também, a curto prazo, fazer uma selecção de problemas de grande dimensão, a partir da colecção CUTE [4], de forma a testar a eficiência do algoritmo pseudo-Newton modificado e truncado com procura curvilínea , com base nos tempos de execução. A optimização do código e a comparação dos resultados obtidos com dois dos pacotes de *software* mais populares, especialmente desenvolvidos para problemas de grandes dimensões, como o LANCELOT ([18]) e o LOQO ([104] e [105]), serão a médio prazo dois objectivos a concretizar.

A extensão, da filosofia subjacente ao algoritmo desenvolvido, a problemas com restrições de desigualdade, será feita posteriormente, através de variáveis de folga. Estas serão tratadas pela reformulação do problema com base em funções barreira logarítmicas ou

funções de penalidade exponenciais. Também se pensa investir algum esforço numa outra fórmula de actualização do parâmetro de penalidade que acompanhe o comportamento do algoritmo ao longo do processo iterativo.

O objectivo final será pois o desenvolvimento de um pacote de *software*, a funcionar nos moldes do NPSOL [51], que será disponibilizado à comunidade científica e que irá servir como ferramenta informática em aulas laboratoriais no domínio da optimização não linear.

"Comecei a viver estudando, e acabei estudando para viver". - Bacon

Apêndice A

Tabelas de Resultados

Nas tabelas que se seguem:

- *et* significa "excesso de tempo";
- *ov* significa "*overflow*";
- *bloq* significa "bloqueio";
- *napd* significa "não atinge precisão definida" ($\|\nabla L\|_2 \leq 10^{-4}$).

Na tabela A.7, para os resultados do NPSOL:

I=1	Qualquer coisa não correu normalmente
I=2	KKT satisfeitas; a F. Mérito não melhora mais
I=3	Não há pontos admissíveis para as rest. lineares
I=4	Não há pontos admissíveis para as rest. não lineares
I=5	Limite de iterações atingido
I=6	KKT não satisfeitas; F. mérito não melhora
I=7	Derivadas da f.o. e das rest. parecem incorrectas

Na tabela A.8, para os resultados do MATLAB:

- *nc* significa "não converge";
- *nhsa* significa "não há solução admissível".

Tabela A.1: Tabela com resultados da versão Linear/Armijo

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	2	2	10^{-13}	10^{-15}	10^{-16}	7
ALJAZZAF	>300			10^{3}	10^{1}	10^{2}	851
ARGTRIG	4	5	5	10^{-6}	10^{-7}	10^{-6}	65
ARTIF	7	8	8	10^{-8}	10^{-8}	10^{-8}	135
AVION2	*et*						
BDVALUE	2	3	3	10^{-11}	10^{-10}	10^{-10}	50
BOOTH	1	2	2	10^{-14}	10^{-14}	10^{-14}	9
BRATU2D	18	115	19	10^{-7}	10^{-6}	10^{-8}	10^{5}
BRATU2DT	24	163	25	10^{-7}	10^{-6}	10^{-6}	1.9×10^{5}
BRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	24
BROYDN3D	4	5	5	10^{-8}	10^{-8}	10^{-8}	63
BROYDNBD	6	7	7	10^{-6}	10^{-7}	10^{-7}	90
BT1	>200			10^{3}	10^{2}	10^{3}	627
BT10	6	8	7	10^{-7}	10^{-8}	-1.0	27
BT11	14	15	15	10^{-6}	10^{-13}	0.825	60
BT12	21	22	22	10^{-6}	10^{-12}	6.188	87
BT13	>500			10^{0}	0.00	-10^{9}	1331
BT2	45	46	46	10^{-6}	10^{-13}	0.3×10^{-1}	160
BT3	1	2	2	10^{-12}	10^{-13}	0.4×10^{1}	9
BT4	>300			10^{-3}	10^{-7}	-0.45×10^{2}	673
BT5	17	19	18	10^{-6}	10^{-11}	0.96×10^{3}	65
BT6	>500			10^{2}	10^{1}	0.23×10^{2}	2644
BT7	*et*						
BT8	141	892	142	10^{-6}	10^{-6}	0.1×10^{1}	741
BT9	>400			10^{0}	10^{-3}	10^{-2}	1535
BYRDSPHR	121	148	122	10^{-6}	10^{-13}	-0.46×10^{1}	432
CBRATU2D	15	133	16	10^{-9}	10^{-8}	-10^{-8}	10^{4}
CBRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	208
CHANDHEQ	>1000			10^{-4}	10^{-8}	-10^{-9}	2×10^{4}
CLUSTER	8	23	9	10^{-6}	10^{-7}	-10^{-7}	43
COOLHANS	11	17	12	10^{-6}	10^{-10}	10^{-10}	144
DECONVC	*et*						
DEGENLPB	*et*						
DIXCHLNG	*et*						
DNIEPER	*et*						
DRUGDISE	*et*						
DTOC1L	8	9	9	10^{-8}	10^{-14}	0.3×10^{-1}	3×10^{4}
DTOC1NA	52	53	52	10^{-6}	10^{-11}	0.32×10^{-1}	1.7×10^{5}
DTOC1NB	*et*						
DTOC1NC	*et*						
DTOC1ND	*et*						
DTOC3	1	2	2	10^{-13}	10^{-13}	-10^{-13}	430
DTOC4	4	5	5	10^{-11}	10^{-13}	-10^{-13}	10^{3}
EIGMAXC	6	8	7	10^{-11}	10^{-11}	-1.0	10^{3}
EIGMINC	6	8	7	10^{-11}	10^{-11}	1.0	10^{3}
FCCU	1	2	2	10^{-12}	10^{-13}	11.15	61
GENHS28	1	2	2	10^{-13}	10^{-14}	0.927	21
GILBERT	> 100						
GRIDNETA	*et*						
HAGER1	1	2	2	10^{-14}	10^{-14}	-10^{-14}	93
HAGER2	1	2	2	10^{-14}	10^{-13}	10^{-13}	93
HAGER3	1	2	2	10^{-13}	10^{-14}	10^{-14}	93
HAGER4	1	2	2	10^{-13}	10^{-14}	-10^{-14}	93
HATFLDG	6	20	7	10^{-14}	10^{-7}	-10^{-13}	4430
HEART6	28	96	29	10^{-8}	10^{-11}	-10^{-11}	154
HEART8	225	2479	226	10^{-8}	10^{-9}	10^{-10}	7697
HIMMELBE	2	3	3	10^{-15}	0.00	0.00	6
HONG	*et*						
HS100LNP	>700			10^{2}	10^{0}	10^{4}	4321
HS104LNP	saiu	do	CUTE				
HS107	>100						
HS111	>1000			10^{0}	10^{-2}	-0.4×10^{2}	1001
HS111LNP	>1000			10^{0}	10^{-2}	-0.4×10^{2}	1001
HS112	8	16	9	10^{-11}	10^{-15}	-0.4×10^{2}	42
HS119	*et*						
HS26	>300			10^{-3}	10^{-7}	0.36×10^{-5}	121
HS27	245	246	246	10^{-6}	10^{-12}	0.4×10^{-1}	96
HS28	1	2	2	10^{-14}	10^{-14}	10^{-14}	5
HS39	>400			10^{0}	10^{0}	0.23×10^{-1}	983

Tabela A.1: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	12	13	13	10^{-6}	10^{-12}	-0.25	14
HS41	112	226	113	10^{-6}	10^{-17}	1.(9)	105
HS42	54	55	55	10^{-6}	10^{-13}	13.86	33
HS46	>500			10^{-1}	10^{-4}	0.38×10^{-2}	1094
HS47	>500			10^{-4}	10^{-7}	10^{-8}	1121
HS48	1	2	2	10^{-14}	10^{-14}	-10^{-15}	7
HS49	16	17	17	10^{-6}	10^{-15}	10^{-8}	16
HS50	9	10	10	10^{-14}	10^{-14}	10^{-15}	13
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	7
HS52	1	2	2	10^{-13}	10^{-14}	5.326	6
HS53	1	2	2	10^{-13}	10^{-14}	4.093	6
HS54	5	7	6	10^{-9}	10^{-12}	-0.9035	9
HS55	149	538	150	10^{-6}	10^{-6}	6.(6)	2499
HS56	>700			10^{-1}	10^{-3}	0.349×10^{-3}	3603
HS6	>200			10^{1}	10^{2}	0.165×10^{4}	304
HS60	35	36	36	10^{-6}	10^{-13}	0.325×10^{-1}	19
HS61	19	34	20	10^{-6}	10^{-13}	-0.143×10^{3}	23
HS62	7	10	8	10^{-10}	10^{-17}	-0.26×10^{5}	11
HS63	17	19	18	10^{-6}	10^{-11}	961.7	14
HS68	>100						
HS69	*et*						
HS7	25	26	26	10^{-6}	10^{-12}	-0.17×10^{1}	14
HS77	>500			10^{1}	10^{-1}	19.36	1646
HS78	8	9	9	10^{-6}	10^{-11}	-2.919	13
HS79	>500			10^{-4}	10^{-8}	-0.788×10^{-1}	1244
HS8	4	6	5	10^{-8}	10^{-7}	-0.(9)	7
HS80	113	114	114	10^{-6}	10^{-12}	0.539×10^{-1}	102
HS81	105	106	106	10^{-6}	10^{-12}	0.539×10^{-1}	97
HS9	3	4	4	10^{-10}	10^{-13}	-0.5	6
HS99	31	47	32	10^{-7}	10^{-12}	-0.83×10^{9}	98
HS99EXP	*et*						
HYDCAR6	5	7	6	10^{-11}	10^{-11}	-0.28×10^{-12}	2961
HYPCIR	4	6	5	10^{-9}	10^{-8}	0.6×10^{-9}	7
INTEGREQ	*et*						
LCH	1	2	2	10^{-9}	10^{-15}	-0.31×10^{1}	2×10^{4}
LIN	*et*						
LOTSCHD	*et*						
LSNNODOC	*et*						
MARATOS	17	18	18	10^{-6}	10^{-12}	-1.0	11
METHANB8	3	4	4	10^{-11}	10^{-12}	-0.16×10^{-11}	2208
METHANL8	4	5	5	10^{-11}	10^{-10}	-0.5×10^{-13}	2771
MINPERM	*et*						
MSQRTA	4	7	5	10^{-6}	10^{-7}	-0.21×10^{-8}	9
MSQRTB	9	29	10	10^{-11}	10^{-11}	-0.33×10^{-11}	154
ODFITS	7	8	8	10^{-8}	10^{-12}	-0.23×10^{4}	43
OPTCNTRL	14	15	15	10^{-6}	10^{-9}	0.35×10^{-8}	4826
ORTHREGB	*et*						
ORTHREGC	*et*						
ORTHREGD	*et*						
ORTHREGE	*et*						
PFIT1	8	22	9	10^{-9}	10^{-10}	-0.7×10^{-10}	20
PFIT2	8	21	9	10^{-12}	10^{-13}	-0.3×10^{-13}	21
PFIT3	7	20	8	10^{-7}	10^{-9}	-0.145×10^{-8}	25
PFIT4	8	20	9	10^{-10}	10^{-13}	0.35×10^{-13}	20
PORTFL1	1	2	2	10^{-14}	10^{-15}	0.187×10^{-1}	13
PORTFL2	1	2	2	10^{-14}	0.00	0.268×10^{-1}	13
PORTFL3	1	2	2	10^{-15}	10^{-15}	0.271×10^{-1}	13
PORTFL4	1	2	2	10^{-19}	10^{-16}	0.212×10^{-1}	13
PORTFL6	1	2	2	10^{-14}	10^{-16}	0.223×10^{-1}	13
POWELLBS	21	93	22	10^{-7}	10^{-13}	-0.22×10^{-13}	29
POWELLSQ	>200			10^{2}	10^{2}	0.137×10^{2}	356
READING5	1	10	2	10^{-11}	10^{-13}	10^{-12}	3.54×10^{2}
RECIPE	>300			10^{0}	10^{-1}	0.184×10^{-1}	857
RK23	*et*						
ROBOT	5	7	6	10^{-6}	10^{-6}	-0.45×10^{-6}	31
SEMICON1	1	2	2	10^{-10}	10^{-12}	-0.46×10^{-12}	31
SEMICON2	1	2	2	10^{-12}	10^{-14}	0.355×10^{-14}	31
TENBARS2	*et*						
TENBARS3	*et*						
TRIGGER	9	41	10	10^{-14}	10^{-12}	0.56×10^{-12}	84
ZANGWIL3	1	2	2	0.00	10^{-13}	0.2×10^{-13}	6

Tabela A.2: Tabela com resultados da versão Linear/CAA

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	3	3	10^{-13}	10^{-15}	10^{-16}	11
ALJAZZAF	>300			10^{3}	10^{0}	0.5×10^{2}	208
ARGTRIG	4	9	9	10^{-8}	10^{-9}	-0.185×10^{-8}	83
ARTIF	11	51	23	10^{-7}	10^{-8}	-0.47×10^{-8}	569
AVION2	*et*						
BDVALUE	2	5	5	10^{-11}	10^{-10}	0.6×10^{-10}	66
BOOTH	1	3	3	10^{-14}	10^{-14}	10^{-14}	6
BRATU2D	*et*						
BRATU2DT	*et*						
BRATU3D	1	3	3	10^{-14}	10^{-15}	10^{-15}	27
BROYDN3D	4	9	9	10^{-8}	10^{-8}	10^{-8}	81
BROYDNBD	627	1256	1255	10^{-8}	10^{-8}	10^{-8}	10^{4}
BT1	>200			10^{7}	1.0	-0.1×10^{3}	128
BT10	11	24	23	10^{-7}	10^{-8}	-1.0	16
BT11	12	25	25	10^{-6}	10^{-11}	0.825	24
BT12	15	31	31	10^{-8}	10^{-8}	0.619	27
BT13	>500			10^{0}	10^{-2}	-0.86×10^{2}	447
BT2	39	79	79	10^{-6}	10^{-10}	0.326×10^{-1}	33
BT3	1	3	3	10^{-12}	10^{-13}	4.093	7
BT4	168	374	337	10^{-6}	10^{-7}	-45.5	146
BT5	13	28	27	10^{-7}	10^{-14}	961.7	20
BT6	>500			10^{0}	10^{-3}	0.322	632
BT7	>500			10^{4}	10^{1}	1052.4	753
BT8	66	391	133	10^{-11}	10^{-12}	1.0	192
BT9	>400			10^{0}	10^{0}	0.758	409
BYRDSPHR	>300			10^{11}	10^{2}	18.0	246
CBRATU2D	7	15	15	10^{-7}	10^{-6}	-0.27×10^{-6}	1459
CBRATU3D	1	3	3	10^{-14}	10^{-15}	10^{-15}	295
CHANDHEQ	>1000			10^{-3}	10^{-7}	-0.65×10^{-6}	3×10^{4}
CLUSTER	7	29	15	10^{-7}	10^{-8}	-0.15×10^{-8}	22
COOLHANS	11	24	23	10^{-6}	10^{-10}	10^{-10}	157
DECONVC	5	11	11	10^{-8}	10^{-12}	10^{-12}	940
DEGENLPB	*et*						
DIXCHLNG	>1000			10^{17}	10^{10}	10^{14}	7817
DNIEPER	*et*						
DRUGDISE	*et*						
DTOC1L	7	15	15	10^{-7}	10^{-14}	0.307×10^{-1}	4×10^{4}
DTOC1NA	11	23	23	10^{-6}	10^{-10}	0.32×10^{-1}	7×10^{4}
DTOC1NB	*et*						
DTOC1NC	*et*						
DTOC1ND	*et*						
DTOC3	1	3	3	10^{-13}	10^{-13}	10^{-13}	628
DTOC4	4	9	9	10^{-11}	10^{-13}	10^{-13}	1892
EIGMAXC	6	13	13	10^{-7}	10^{-7}	-1.0	1903
EIGMINC	6	13	13	10^{-7}	10^{-7}	1.0	1902
FCCU	1	3	3	10^{-12}	10^{-13}	11.15	81
GENHS28	1	3	3	10^{-13}	10^{-14}	0.927	24
GILBERT	20	56	41	10^{-6}	10^{-10}	21.15	2266
GRIDNETA	*et*						
HAGER1	1	3	3	10^{-14}	10^{-14}	10^{-14}	134
HAGER2	1	3	3	10^{-13}	10^{-14}	10^{-13}	134
HAGER3	1	3	3	10^{-13}	10^{-14}	10^{-14}	135
HAGER4	1	3	3	10^{-13}	10^{-14}	10^{-14}	134
HATFLDG	*et*						
HEART6	>600			10^{4}	10^{2}	0.67×10^{2}	2345
HEART8	>800			10^{3}	10^{2}	0.23×10^{1}	6867
HIMMELBE	2	5	5	10^{-15}	0.00	0.00	7
HONG	*et*						
HS100LNP	>700			10^{2}	10^{2}	731.0	1203
HS104LNP	saiu	do	CUTE				
HS107	>900			10^{14}	10^{12}	10^{19}	6901
HS111	>1000			10^{0}	10^{-2}	-41.76	10^{4}
HS111LNP	>1000			10^{0}	10^{-2}	-41.76	10^{4}
HS112	8	24	17	10^{-6}	10^{-15}	-47.7	63
HS119	*et*						
HS26	>300			10^{-3}	10^{-9}	0.1×10^{-5}	216
HS27	>300			10^{-3}	10^{-4}	0.399×10^{-1}	205
HS28	1	3	3	10^{-14}	10^{-14}	10^{-14}	6
HS39	>400			10^{0}	10^{0}	0.758	408

Tabela A.2: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	7	15	15	10^{-6}	10^{-6}	-0.25	14
HS41	>100						
HS42	9	19	19	10^{-6}	10^{-6}	13.86	14
HS46	38	105	77	10^{-6}	10^{-9}	3.95	64
HS47	187	375	375	10^{-14}	10^{-14}	10^{-8}	300
HS48	1	3	3	10^{-14}	10^{-14}	10^{-15}	7
HS49	13	27	27	10^{-6}	10^{-15}	10^{-11}	21
HS50	7	15	15	10^{-8}	10^{-14}	10^{-15}	16
HS51	1	3	3	10^{-14}	10^{-14}	10^{-14}	7
HS52	12	87	25	10^{-13}	10^{-14}	5.326	51
HS53	1	3	3	10^{-13}	10^{-14}	4.093	7
HS54	5	11	11	10^{-9}	10^{-12}	-0.9035	11
HS55	>600			10^{-4}	10^{-4}	6.66	8654
HS56	>700			10^{-1}	10^{-4}	-0.14×10^{-2}	2272
HS6	>200			10^{1}	10^{1}	45.58	207
HS60	28	57	57	10^{-6}	10^{-12}	0.325×10^{-1}	25
HS61	13	37	27	10^{-6}	10^{-12}	-81.91	27
HS62	6	14	13	10^{-7}	10^{-16}	-0.26×10^{5}	10
HS63	13	28	27	10^{-7}	10^{-14}	961.7	20
HS68	>1000						
HS69	*et*						
HS7	15	31	31	10^{-6}	10^{-7}	-1.732	15
HS77	>500			10^{3}	10^{1}	406.6	619
HS78	7	15	15	10^{-6}	10^{-10}	-2.919	17
HS79	21	43	43	10^{-6}	10^{-11}	0.78×10^{-1}	37
HS8	5	11	11	10^{-7}	10^{-7}	-0.(9)	9
HS80	13	27	27	10^{-6}	10^{-9}	0.539×10^{-1}	25
HS81	11	23	23	10^{-6}	10^{-9}	0.539×10^{-1}	23
HS9	3	7	7	10^{-10}	10^{-13}	-0.5	7
HS99	>700			10^{-7}	10^{-10}	-0.83×10^{9}	3244
HS99EXP	*et*						
HYDCAR6	6	13	13	10^{-11}	10^{-12}	10^{-12}	5422
HYPCIR	5	11	11	10^{-8}	10^{-7}	0.16×10^{-7}	9
INTEGREQ	*et*						
LCH	1	3	3	10^{-9}	10^{-15}	-0.3099×10^{1}	2.8×10^{4}
LIN	*et*						
LOTSCHD	*et*						
LSNNODOC	*et*						
MARATOS	9	19	19	10^{-6}	10^{-6}	-1.0	11
METHANB8	3	7	7	10^{-11}	10^{-12}	10^{-11}	3758
METHANL8	4	9	9	10^{-11}	10^{-10}	10^{-13}	4835
MINPERM	*et*						
MSQRTA	19	39	39	10^{-8}	10^{-9}	10^{-10}	33
MSQRTB	55	111	111	10^{-10}	10^{-10}	10^{-10}	685
ODFITS	6	13	13	10^{-8}	10^{-13}	-0.238×10^{4}	62
OPTCNTRL	13	27	27	10^{-6}	10^{-6}	-0.18×10^{-6}	8.346×10^{3}
ORTHREGB	*et*						
ORTHREGC	*et*						
ORTHREGD	*et*						
ORTHREGE	*et*						
PFIT1	>300			10^{8}	10^{8}	10^{8}	361
PFIT2	>300			10^{11}	10^{8}	10^{8}	363
PFIT3	*et*						
PFIT4	>300			10^{12}	10^{11}	-10^{12}	359
PORTFL1	1	3	3	10^{-14}	10^{-15}	0.187×10^{-1}	17
PORTFL2	1	3	3	10^{-14}	0.00	0.27×10^{-1}	17
PORTFL3	1	3	3	10^{-15}	10^{-15}	0.27×10^{-1}	18
PORTFL4	1	3	3	10^{-15}	10^{-16}	0.212×10^{-1}	17
PORTFL6	1	3	3	10^{-14}	10^{-16}	0.22×10^{-1}	17
POWELLBS	142	454	285	10^{-15}	10^{-8}	10^{-15}	140
POWELLSQ	>200			10^{3}	10^{1}	10.85	144
READING5	1	3	3	10^{-12}	10^{-14}	10^{-14}	1.1×10^{4}
RECIPE	60	242	121	10^{-6}	10^{-8}	10^{-8}	166
RK23	>1700			10^{2}	10^{2}	-10^{-5}	10^{5}
ROBOT	5	11	11	10^{-8}	10^{-8}	10^{-8}	43
SEMICON1	1	3	3	10^{-10}	10^{-12}	10^{-12}	42
SEMICON2	1	3	3	10^{-12}	10^{-14}	10^{-14}	42
TENBARS2	*et*						
TENBARS3	*et*						
TRIGGER	9	45	19	10^{-14}	10^{-13}	10^{-13}	95
ZANGWIL3	1	3	3	0.0	10^{-13}	10^{-13}	7

Tabela A.3: Tabela com resultados da versão Linear/NMonótona

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	2	2	10^{-13}	10^{-15}	10^{-16}	9
ALJAZZAF	>300			10^{3}	10^{1}	50.09	309
ARGTRIG	4	5	5	10^{-6}	10^{-7}	-10^{-6}	51
ARTIF	7	8	8	10^{-8}	10^{-8}	-10^{-8}	112
AVION2	*et*						
BDVALUE	2	3	3	10^{-11}	10^{-10}	10^{-10}	43
BOOTH	1	2	2	10^{-14}	10^{-14}	10^{-14}	5
BRATU2D	14	15	15	10^{-8}	10^{-6}	10^{-6}	6.7×10^{4}
BRATU2DT	21	52	22	10^{-8}	10^{-7}	-10^{-7}	6.5×10^{4}
BRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	20
BROYDN3D	4	5	5	10^{-8}	10^{-8}	-10^{-8}	51
BROYDNBD	6	7	7	10^{-6}	10^{-7}	10^{-7}	70
BT1	>200			10^{2}	10^{1}	0.67×10^{3}	112
BT10	6	7	7	10^{-7}	10^{-8}	-1.0	7
BT11	14	15	15	10^{-6}	10^{-13}	0.825	17
BT12	21	22	22	10^{-6}	10^{-12}	6.188	22
BT13	>500			10^{0}	0.00	-10^{-12}	1432
BT2	45	46	46	10^{-6}	10^{-13}	$0.325\times y\ 10^{-1}$	20
BT3	1	2	2	10^{-12}	10^{-13}	4.093	6
BT4	>300			10^{-1}	10^{-2}	-0.45×10^{2}	178
BT5	17	19	18	10^{-6}	10^{-11}	961.7	12
BT6	>500			10^{2}	10^{2}	28.65	1878
BT7	*et*						
BT8	160	693	161	10^{-7}	10^{-7}	1.0	273
BT9	>400			10^{0}	10^{0}	-1.739	449
BYRDSPHR	121	148	122	10^{-6}	10^{-13}	-4.683	63
CBRATU2D	18	36	19	10^{-7}	10^{-6}	-10^{-6}	3449
CBRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	208
CHANDHEQ	>1000			10^{-4}	10^{-9}	-10^{-8}	2×10^{4}
CLUSTER	8	23	9	10^{-6}	10^{-7}	-10^{-7}	19
COOLHANS	11	17	12	10^{-6}	10^{-10}	10^{-10}	112
DECONVC	>6100			10^{37}	10^{35}	10^{45}	6×10^{6}
DEGENLPB	>2000			10^{3}	10^{1}	10^{7}	2.5×10^{6}
DIXCHLNG	>1000			10^{14}	10^{3}	10^{16}	1.1×10^{5}
DNIEPER	*et*						
DRUGDISE	*et*						
DTOC1L	8	9	9	10^{-8}	10^{-14}	0.307×10^{-1}	2.9×10^{4}
DTOC1NA	52	53	53	10^{-6}	10^{-11}	0.32×10^{-1}	1.7×10^{5}
DTOC1NB	*et*						
DTOC1NC	*et*						
DTOC1ND	*et*						
DTOC3	1	2	2	10^{-13}	10^{-13}	-10^{-13}	430
DTOC4	4	5	5	10^{-11}	10^{-13}	-10^{-13}	1102
EIGMAXC	6	8	7	10^{-11}	10^{-11}	-1.0	1216
EIGMINC	6	8	7	10^{-11}	10^{-11}	1.0	1216
FCCU	1	2	2	10^{-12}	10^{-13}	11.150	57
GENHS28	1	2	2	10^{-13}	10^{-14}	0.927	17
GILBERT	>5000			10^{-3}	10^{-8}	21.153	3.9×10^{5}
GRIDNETA	*et*						
HAGER1	1	2	2	10^{-14}	10^{-14}	-10^{-14}	93
HAGER2	1	2	2	10^{-13}	10^{-14}	10^{-13}	93
HAGER3	1	2	2	10^{-13}	10^{-14}	10^{-14}	93
HAGER4	1	2	2	10^{-13}	10^{-14}	-10^{-14}	93
HATFLDG	6	20	7	10^{-14}	10^{-7}	-10^{-13}	4473
HEART6	18	47	19	10^{-10}	10^{-13}	-10^{-13}	85
HEART8	51	215	52	10^{-9}	10^{-10}	10^{-10}	740
HIMMELBE	2	3	3	10^{-15}	0.00	0.00	6
HONG	*et*						
HS100LNP	>700			10^{2}	10^{-2}	841.04	5429
HS104LNP	saiu	do	CUTE				
HS107	>900			10^{19}	10^{12}	-10^{27}	1.7×10^{5}
HS111	*et*						
HS111LNP	*et*						
HS112	8	16	9	10^{-11}	10^{-15}	-47.76	51
HS119	*et*						
HS26	>300			10^{-3}	10^{-7}	0.358×10^{-5}	118
HS27	245	248	246	10^{-6}	10^{-12}	0.04	95
HS28	1	2	2	10^{-14}	10^{-14}	10^{-14}	5
HS39	>400			10^{0}	10^{0}	-0.6	466

Tabela A.3: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	12	13	13	10^{-6}	10^{-12}	-0.25	13
HS41	313	1152	314	10^{-6}	10^{-15}	2.0	306
HS42	54	55	55	10^{-6}	10^{-13}	13.858	32
HS46	>500			10^{-3}	10^{-4}	-0.9×10^{-4}	874
HS47	>500			10^{-4}	10^{-8}	10^{-8}	760
HS48	1	2	2	10^{-14}	10^{-14}	-10^{-15}	6
HS49	16	17	17	10^{-6}	10^{-15}	10^{-8}	16
HS50	9	10	10	10^{-14}	10^{-14}	10^{-15}	13
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	7
HS52	1	2	2	10^{-13}	10^{-14}	5.327	7
HS53	1	2	2	10^{-13}	10^{-14}	4.093	6
HS54	5	7	6	10^{-9}	10^{-12}	-0.9035	8
HS55	116	228	117	10^{-6}	10^{-6}	6.(6)	1323
HS56	>700			10^{-2}	10^{-3}	-0.429×10^{-3}	1679
HS6	>200			10^{1}	10^{-1}	5.411	311
HS60	35	36	36	10^{-6}	10^{-13}	0.356×10^{-1}	18
HS61	19	34	20	10^{-6}	10^{-13}	-143.64	22
HS62	7	9	8	10^{-10}	10^{-17}	-0.262×10^{5}	8
HS63	17	19	18	10^{-6}	10^{-11}	961.72	12
HS68	>400			10^{0}	10^{-2}	-0.45	753
HS69	*et*						
HS7	25	26	26	10^{-6}	10^{-12}	-1.732	14
HS77	>500			10^{1}	10^{-2}	4.073	1889
HS78	8	9	9	10^{-6}	10^{-11}	-2.920	12
HS79	>500			10^{-4}	10^{-8}	0.788×10^{-1}	635
HS8	4	6	5	10^{-8}	10^{-7}	-0.(9)	7
HS80	113	114	114	10^{-6}	10^{-12}	0.539×10^{-1}	100
HS81	105	106	106	10^{-6}	10^{-12}	0.539×10^{-1}	95
HS9	3	4	4	10^{-10}	10^{-13}	-0.5	6
HS99	31	41	32	10^{-7}	10^{-12}	-0.83×10^{9}	74
HS99EXP	*et*						
HYDCAR6	5	6	6	10^{-11}	10^{-11}	-10^{-12}	2999
HYPCIR	4	6	5	10^{-9}	10^{-8}	10^{-9}	7
INTEGREQ	*et*						
LCH	1	2	2	10^{-9}	10^{-15}	-3.099	19893
LIN	*et*						
LOTSCHD	*et*						
LSNNODOC	*et*						
MARATOS	17	18	18	10^{-6}	10^{-12}	-1.0	11
METHANB8	3	4	4	10^{-11}	10^{-12}	-10^{-11}	2239
METHANL8	4	5	5	10^{-11}	10^{-10}	-10^{-13}	2808
MINPERM	*et*						
MSQRTA	4	7	5	10^{-6}	10^{-7}	-10^{-8}	9
MSQRTB	9	29	10	10^{-11}	10^{-11}	-10^{-11}	155
ODFITS	7	8	8	10^{-8}	10^{-12}	-0.238×10^{4}	43
OPTCNTRL	14	15	15	10^{-6}	10^{-9}	10^{-8}	4883
ORTHREGB	*et*						
ORTHREGC	*et*						
ORTHREGD	*et*						
ORTHREGE	*et*						
PFIT1	8	22	9	10^{-9}	10^{-10}	-10^{-10}	20
PFIT2	8	21	9	10^{-12}	10^{-13}	-10^{-13}	20
PFIT3	7	20	8	10^{-7}	10^{-9}	-10^{-8}	25
PFIT4	8	20	9	10^{-10}	10^{-13}	10^{-13}	19
PORTFL1	1	2	2	10^{-14}	10^{-15}	0.1869×10^{-1}	12
PORTFL2	1	2	2	10^{-14}	0.00	0.268×10^{-1}	13
PORTFL3	1	2	2	10^{-15}	10^{-15}	0.271×10^{-1}	12
PORTFL4	1	2	2	10^{-15}	10^{-16}	0.212×10^{-1}	13
PORTFL6	1	2	2	10^{-14}	10^{-16}	0.223×10^{-1}	12
POWELLBS	13	14	14	10^{-8}	10^{-15}	-10^{-14}	10
POWELLSQ	>200			10^{0}	10^{-1}	0.1133	412
READING5	1	10	2	10^{-11}	10^{-13}	10^{-12}	3.6×10^{4}
RECIPE	>300			10^{0}	10^{-1}	0.319×10^{-1}	405
RK23	*et*						
ROBOT	5	6	6	10^{-6}	10^{-6}	-0.45×10^{-6}	30
SEMICON1	1	2	2	10^{-10}	10^{-12}	-10^{-12}	30
SEMICON2	1	2	2	10^{-12}	10^{-14}	10^{-14}	30
TENBARS2	*et*						
TENBARS3	*et*						
TRIGGER	10	36	11	10^{-7}	10^{-9}	-10^{-9}	74
ZANGWIL3	1	2	2	0.00	10^{-13}	10^{-13}	6

Tabela A.4: Tabela com resultados da versão Curvilínea com PCCAA

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	2	2	10^{-13}	10^{-15}	10^{-16}	13
ALJAZZAF	>300		612	10^{11}	10^{18}	10^{18}	613
ARGTRIG	4	8	8	10^{-10}	10^{-11}	-10^{-11}	79
ARTIF	5	11	11	10^{-8}	10^{-8}	10^{-8}	145
AVION2	*et*						
BDVALUE	1	3	3	10^{-8}	10^{-7}	-0.12×10^{-6}	45
BOOTH	1	2	2	10^{-14}	10^{-14}	10^{-14}	9
BRATU2D	3	22	7	10^{-11}	10^{-10}	-10^{-10}	2.6×10^{4}
BRATU2DT	3	4	4	10^{-13}	10^{-13}	-0.1×10^{-12}	5099
BRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	24
BROYDN3D	3	7	7	10^{-14}	10^{-14}	-10^{-13}	67
BROYDNBD	4	9	9	10^{-9}	10^{-9}	10^{-10}	86
BT1	>200		401	10^{2}	10^{1}	136.13	563
BT10	4	5	5	10^{-6}	10^{-7}	-1.0	9
BT11	9	18	18	10^{-6}	10^{-13}	0.825	22
BT12	8	16	16	10^{-6}	10^{-12}	6.188	21
BT13	>500		1001	10^{1}	10^{-1}	0.27×10^{-1}	1035
BT2	>300		601	10^{4}	10^{6}	10^{7}	229
BT3	1	2	2	10^{-12}	10^{-13}	0.409×10^{1}	10
BT4	25	26	26	10^{-6}	10^{-12}	-3.705	29
BT5	308	616	309	10^{-6}	10^{-11}	952.14	275
BT6	>500						
BT7	*ov*						
BT8	12	13	13	10^{-9}	10^{-10}	1.0	36
BT9	73	74	74	10^{-6}	10^{-8}	-1.0	107
BYRDSPHR	25	52	51	10^{-8}	10^{-8}	-4.683	34
CBRATU2D	6	13	13	10^{-14}	10^{-15}	-10^{-15}	1254
CBRATU3D	1	2	2	10^{-15}	10^{-14}	10^{-15}	207
CHANDHEQ	>1000		2001	10^{-4}	10^{-9}	-10^{-8}	9×10^{4}
CLUSTER	7	8	8	10^{-7}	10^{-8}	-0.156×10^{-8}	30
COOLHANS	21	117	42	10^{-6}	10^{-10}	10^{-10}	672
DECONVC	4	8	8	10^{-8}	10^{-14}	10^{-14}	702
DEGENLPB	*et*						
DIXCHLNG	*et*						
DNIEPER	*et*						
DRUGDISE	*et*						
DTOC1L	8	9	9	10^{-8}	10^{-14}	0.307×10^{-1}	2.88×10^{4}
DTOC1NA	22	31	31	10^{-11}	10^{-6}	0.321×10^{-1}	9.4×10^{4}
DTOC1NB	*et*						
DTOC1NC	*et*						
DTOC1ND	*et*						
DTOC3	1	2	2	10^{-13}	10^{-13}	-10^{-13}	428
DTOC4	3	7	7	10^{-7}	10^{-6}	10^{-7}	1.46×10^{3}
EIGMAXC	7	8	8	10^{-7}	10^{-8}	-0.(9)	1235
EIGMINC	7	8	8	10^{-7}	10^{-8}	1.0	1233
FCCU	1	2	2	10^{-13}	10^{-12}	11.149	61
GENHS28	1	2	2	10^{-14}	10^{-13}	0.927	21
GILBERT	20	56	39	10^{-10}	10^{-6}	21.15	2.29×10^{3}
GRIDNETA	*et*						
HAGER1	1	2	2	10^{-14}	10^{-14}	-10^{-14}	95
HAGER2	1	2	2	10^{-14}	10^{-13}	10^{-13}	95
HAGER3	1	2	2	10^{-14}	10^{-13}	10^{-14}	95
HAGER4	1	2	2	10^{-14}	10^{-13}	-10^{-14}	96
HATFLDG	813	1627	1627	10^{-14}	10^{-9}	10^{-13}	3.8×10^{5}
HEART6	*ov*						
HEART8	*ov*						
HIMMELBE	1	3	3	10^{-15}	10^{-15}	-10^{-15}	10
HONG	16	17	17	10^{-6}	10^{-15}	22.571	20
HS100LNP	*ov*						
HS104LNP	10	11	11	10^{-6}	10^{-12}	3.951	36
HS107	>900		1801	10^{9}	10^{6}	10^{12}	6.8×10^{3}
HS111	>1000						
HS111LNP	>1000						
HS112	8	16	9	10^{-11}	10^{-15}	-47.76	49
HS119	21	22	22	10^{-7}	10^{-14}	200.746	449
HS26	>300						
HS27	19	38	38	10^{-6}	10^{-13}	0.4×10^{-1}	21
HS28	1	2	2	10^{-14}	10^{-14}	10^{-14}	8
HS39	84	85	85	10^{-6}	10^{-7}	-0.(9)	122

Tabela A.4: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	4	9	9	10^{-6}	10^{-6}	-0.25	13
HS41	167	242	220	10^{-6}	10^{-15}	2.0	541
HS42	4	9	9	10^{-6}	10^{-8}	13.858	12
HS46	>500		993	10^{-4}	10^{-10}	0.42×10^{-6}	1666
HS47	647	1153		10^{-6}	10^{-12}	10^{-10}	935
HS48	1	2	2	10^{-14}	10^{-14}	10^{-15}	8
HS49	16	17	17	10^{-6}	10^{-15}	0.1375×10^{-8}	20
HS50	9	10	10	10^{-14}	10^{-14}	10^{-15}	16
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	9
HS52	1	2	2	10^{-13}	10^{-14}	5.327	8
HS53	1	2	2	10^{-13}	10^{-14}	4.093	9
HS54	5	9	9	10^{-9}	10^{-11}	-0.9035	11
HS55	3	22	7	10^{-14}	10^{-15}	6.666	198
HS56	>700		1074	10^{-4}	10^{-14}	-0.335×10^{-6}	3309
HS6	>200		401	10^{1}	10^{9}	10^{24}	620
HS60	25	42	42	10^{-6}	10^{-13}	0.326×10^{-1}	24
HS61	9	19	18	10^{-7}	10^{-14}	-81.92	21
HS62	9	30	10	10^{-11}	10^{-16}	-0.262×10^{5}	30
HS63	>300		601	10^{1}	10^{2}	946.5	269
HS68	>400		801	10^{0}	10^{-2}	-0.4475	1268
HS69	3	22	7	10^{-6}	0.00	-0.315×10^{-3}	21
HS7	13	22	22	10^{-6}	10^{-13}	-1.732	15
HS77	*ov*						
HS78	6	11	11	10^{-6}	10^{-12}	-2.9197	15
HS79	32	83	60	10^{-6}	10^{-12}	0.788×10^{-1}	77
HS8	7	8	8	10^{-8}	10^{-8}	-0.(9)	10
HS80	8	17	17	10^{-7}	10^{-9}	0.539×10^{-1}	21
HS81	9	15	15	10^{-6}	10^{-12}	0.539×10^{-1}	20
HS9	3	5	5	10^{-10}	0.00	-0.5	9
HS99	30	31	31	10^{-7}	0.00	-0.831×10^{9}	42
HS99EXP	*et*						
HYDCAR6	*et*						
HYPCIR	4	9	9	10^{-9}	10^{-9}	-0.12×10^{-8}	10
INTEGREQ	3	7	7	10^{-7}	10^{-7}	-0.23×10^{-6}	2.7×10^{4}
LCH	1	2	2	10^{-9}	10^{-15}	-3.0989	2×10^{4}
LIN	*ov*						
LOTSCHD	*et*						
LSNNODOC	*ov*						
MARATOS	5	9	9	10^{-6}	10^{-12}	-1.0	11
METHANB8	2	4	4	10^{-10}	10^{-10}	-0.18×10^{-10}	2167
METHANL8	3	6	6	10^{-11}	10^{-10}	-0.6×10^{-11}	3246
MINPERM	*et*						
MSQRTA	6	7	7	10^{-10}	10^{-11}	-10^{-12}	12
MSQRTB	*ov*						
ODFITS	7	8	8	10^{-8}	10^{-12}	-0.23×10^{4}	47
OPTCNTRL	10	20	20	10^{-6}	10^{-10}	0.32×10^{-9}	6213
ORTHREGB	*et*						
ORTHREGC	>2500						
ORTHREGD	>2500						
ORTHREGE	*et*						
PFIT1	*ov*						
PFIT2	*ov*						
PFIT3	*ov*						
PFIT4	*ov*						
PORTFL1	1	2	2	10^{-14}	10^{-15}	0.187×10^{-1}	15
PORTFL2	1	2	2	10^{-14}	0.00	0.2683×10^{-1}	15
PORTFL3	1	2	2	10^{-15}	10^{-15}	0.271×10^{-1}	15
PORTFL4	1	2	2	10^{-15}	10^{-16}	0.212×10^{-1}	15
PORTFL6	1	2	2	10^{-14}	10^{-16}	0.223×10^{-1}	16
POWELLBS	9	10	10	10^{-6}	10^{-11}	-10^{-15}	11
POWELLSQ	>200		401	10^{2}	10^{2}	17.865	152
READING5	1	17	2	10^{-11}	10^{-13}	0.2×10^{-12}	5.9×10^{4}
RECIPE	16	17	17	10^{-6}	10^{-12}	0.77×10^{-12}	77
RK23	*et*						
ROBOT	4	8	8	10^{-10}	10^{-10}	-0.4×10^{-10}	36
SEMICON1	1	2	2	10^{-10}	10^{-12}	-0.42×10^{-12}	33
SEMICON2	1	2	2	10^{-12}	10^{-14}	10^{-14}	33
TENBARS2	*et*						
TENBARS3	*et*						
TRIGGER	5	12	11	10^{-4}	10^{-8}	0.127×10^{-7}	40
ZANGWIL3	1	2	2	0.00	10^{-13}	0.2×10^{-13}	8

Tabela A.5: Tabela com resultados da versão Curvilínea com Armijo

Problema	nit	nf	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	2	10^{-13}	10^{-15}	10^{-16}	13
ALJAZZAF	>300		10^{11}	10^{18}	10^{18}	613
ARGTRIG	4	5	10^{-12}	10^{-12}	-10^{-12}	58
ARTIF	6	7	10^{-10}	10^{-11}	10^{-11}	117
AVION2	*et*					
BDVALUE	1	2	10^{-8}	10^{-7}	-10^{-6}	34
BOOTH	1	2	10^{-14}	10^{-14}	10^{-14}	9
BRATU2D	4	15	10^{-13}	10^{-12}	10^{-12}	1.8×10^{4}
BRATU2DT	3	4	10^{-13}	10^{-13}	-0.1×10^{-12}	5099
BRATU3D	1	2	10^{-14}	10^{-15}	10^{-15}	24
BROYDN3D	3	4	10^{-12}	10^{-12}	10^{-12}	48
BROYDNBD	5	6	10^{-13}	10^{-14}	10^{-14}	70
BT1	3130	3538	10^{-6}	10^{-15}	-1.0	1.5×10^{3}
BT10	4	5	10^{-6}	10^{-7}	-1.0	9
BT11	18	19	10^{-6}	10^{-13}	0.825	26
BT12	20	21	10^{-6}	10^{-13}	6.188	28
BT13	52	227	0.0	0.0	5×10^{5}	122
BT2	82	119	10^{-6}	10^{-12}	2.189	57
BT3	1	2	10^{-12}	10^{-13}	0.409×10^{1}	10
BT4	40	84	10^{-6}	10^{-11}	-45.5	46
BT5	18	22	10^{-6}	10^{-11}	961.7	19
BT6	376	1389	10^{-6}	10^{-12}	0.277	1.4×10^{3}
BT7	*ov*					
BT8	12	13	10^{-9}	10^{-10}	1.0	36
BT9	73	74	10^{-6}	10^{-8}	-1.0	107
BYRDSPHR	127	143	10^{-6}	10^{-12}	-4.683	89
CBRATU2D	3	4	10^{-14}	10^{-15}	10^{-16}	427
CBRATU3D	1	2	10^{-15}	10^{-14}	10^{-15}	207
CHANDHEQ	>1000		10^{-3}	10^{-9}	10^{-9}	6×10^{4}
CLUSTER	7	8	10^{-7}	10^{-8}	-0.156×10^{-8}	30
COOLHANS	56	402	10^{-5}	10^{-10}	10^{-10}	2186
DECONVC	7	23	10^{-6}	10^{-15}	10^{-11}	1910
DEGENLPB	*et*					
DIXCHLNG	73	360	10^{-6}	10^{-15}	2.4×10^{3}	10^{3}
DNIEPER	*et*					
DRUGDISE	*et*					
DTOC1L	8	9	10^{-8}	10^{-14}	0.307×10^{-1}	2.88×10^{4}
DTOC1NA	52	53	10^{-6}	10^{-11}	0.321×10^{-1}	1.7×10^{5}
DTOC1NB	*et*					
DTOC1NC	*et*					
DTOC1ND	*et*					
DTOC3	1	2	10^{-13}	10^{-13}	-10^{-13}	429
DTOC4	3	4	10^{-6}	10^{-15}	10^{-12}	912
EIGMAXC	4	7	10^{-11}	10^{-11}	-1.0	103
EIGMINC	4	7	10^{-11}	10^{-11}	1.0	10^{3}
FCCU	1	2	10^{-13}	10^{-12}	11.149	61
GENHS28	1	2	10^{-14}	10^{-13}	0.927	21
GILBERT	54	104	10^{-6}	10^{-12}	21.15	4.5×10^{3}
GRIDNETA	*et*					
HAGER1	1	2	10^{-14}	10^{-14}	-10^{-14}	95
HAGER2	1	2	10^{-14}	10^{-13}	10^{-13}	95
HAGER3	1	2	10^{-14}	10^{-13}	10^{-14}	95
HAGER4	1	2	10^{-14}	10^{-13}	-10^{-14}	96
HATFLDG	18	128	10^{-14}	10^{-8}	10^{-14}	2.8×10^{4}
HEART6	*ov*					
HEART8	*ov*					
HIMMELBE	1	2	10^{-15}	10^{-15}	-10^{-15}	10
HONG	16	17	10^{-6}	10^{-15}	22.571	22
HS100LNP	*ov*					
HS104LNP	Saiu	do	CUTE			
HS107	*ov*					
HS111	>1000					
HS111LNP	>1000					
HS112	8	16	10^{-11}	10^{-15}	-47.76	49
HS119	*ov*					
HS26	722 (*napd*)	726	0.9×10^{-4}	10^{-14}	10^{-7}	393
HS27	243	248	10^{-6}	10^{-12}	0.04	130
HS28	1	2	10^{-14}	10^{-14}	10^{-14}	10
HS39	258	1137	10^{-6}	10^{-12}	-1.0	10^{3}

Tabela A.5: Continuação

Problema	nit	nf	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	12	13	10^{-6}	10^{-12}	-0.25	18
HS41	168	258	10^{-6}	10^{-15}	2.0	467
HS42	54	55	10^{-6}	10^{-13}	13.85	44
HS46	>2000					
HS47	1045	2498	10^{-6}	10^{-11}	10^{-9}	1.8×10^{3}
HS48	1	2	10^{-14}	10^{-14}	10^{-16}	8
HS49	16	17	10^{-6}	10^{-15}	10^{-9}	27
HS50	9	10	10^{-14}	10^{-14}	10^{-16}	16
HS51	1	2	10^{-14}	10^{-14}	10^{-15}	12
HS52	1	2	10^{-13}	10^{-14}	5.327	8
HS53	1	2	10^{-13}	10^{-14}	4.093	9
HS54	5	7	10^{-9}	10^{-12}	-0.9035	12
HS55	2	18	10^{-11}	10^{-10}	6.666	141
HS56	27	44	10^{-6}	10^{-13}	-3.456	92
HS6	798	5947	10^{-6}	0.00	10^{-11}	2.8×10^{3}
HS60	38	39	10^{-6}	10^{-13}	0.032	28
HS61	17	25	10^{-6}	10^{-12}	-81.91	24
HS62	9	30	10^{-11}	10^{-16}	-0.262×10^{5}	40
HS63	18	22	10^{-6}	10^{-11}	961.7	19
HS68	*ov*					
HS69	10	29	10^{-8}	10^{-10}	-986.22	30
HS7	24	25	10^{-6}	10^{-12}	-1.732	19
HS77	>1000					
HS78	6	9	10^{-6}	10^{-11}	-2.919	16
HS79	29	52	10^{-6}	10^{-11}	0.0788	50
HS8	6	10	10^{-8}	10^{-7}	-1.0	12
HS80	112	113	10^{-6}	10^{-12}	0.0539	125
HS81	103	104	10^{-6}	10^{-12}	0.0539	116
HS9	3	4	10^{-10}	0.00	-0.5	9
HS99	33	233	10^{-7}	10^{-12}	-0.83×10^{9}	192
HS99EXP	*et*					
HYDCAR6	5	8	10^{-13}	10^{-13}	10^{-13}	3.4×10^{3}
HYPCIR	4	6	10^{-9}	10^{-8}	-0.17×10^{-8}	10
INTEGREQ	3	4	10^{-7}	10^{-7}	10^{-6}	1.6×10^{4}
LCH	1	2	10^{-9}	10^{-15}	-3.0989	2×10^{4}
LIN	*ov*					
LOTSCHD	*et*					
LSNNODOC	*et*					
MARATOS	17	18	10^{-6}	10^{-12}	-1.0	15
METHANB8	2	3	10^{-10}	10^{-10}	10^{-10}	2100
METHANL8	3	4	10^{-10}	10^{-10}	10^{-11}	3345
MINPERM	*et*					
MSQRTA	7	14	10^{-10}	10^{-11}	10^{-11}	14
MSQRTB	*ov*					
ODFITS	7	8	10^{-8}	10^{-12}	-0.23×10^{4}	56
OPTCNTRL	10	20	10^{-6}	10^{-10}	10^{-10}	6521
ORTHREGB	*et*					
ORTHREGC	>2500					
ORTHREGD	>2500					
ORTHREGE	*et*					
PFIT1	*ov*					
PFIT2	*ov*					
PFIT3	*ov*					
PFIT4	*ov*					
PORTFL1	1	2	10^{-14}	10^{-15}	0.187×10^{-1}	15
PORTFL2	1	2	10^{-14}	0.00	0.2683×10^{-1}	15
PORTFL3	1	2	10^{-15}	10^{-15}	0.271×10^{-1}	15
PORTFL4	1	2	10^{-15}	10^{-16}	0.212×10^{-1}	15
PORTFL6	1	2	10^{-14}	10^{-16}	0.223×10^{-1}	15
POWELLBS	8	10	10^{-6}	10^{-11}	10^{-11}	12
POWELLSQ	>200					
READING5	1	17	10^{-11}	10^{-13}	10^{-12}	5.9×10^{4}
RECIPE	21	22	10^{-6}	10^{-12}	10^{-12}	108
RK23	*et*					
ROBOT	4	5	10^{-9}	10^{-10}	10^{-10}	30
SEMICON1	1	2	10^{-10}	10^{-12}	10^{-13}	33
SEMICON2	1	2	10^{-12}	10^{-14}	10^{-15}	33
TENBARS2	*et*					
TENBARS3	*et*					
TRIGGER	5	21	10^{-7}	10^{-10}	10^{-10}	50
ZANGWIL3	1	2	0.00	10^{-13}	10^{-14}	10

Tabela A.6: Tabela com resultados da versão Curvilínea/NMonótona

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	1	2	2	10^{-13}	10^{-15}	10^{-16}	12
ALJAZZAF	>300		601	10^{11}	10^{18}	10^{18}	565
ARGTRIG	4	8	8	10^{-10}	10^{-11}	-10^{-11}	129
ARTIF	5	11	11	10^{-8}	10^{-8}	10^{-8}	142
AVION2	*et*						
BDVALUE	1	3	3	10^{-8}	10^{-7}	-0.12×10^{-6}	44
BOOTH	1	2	2	10^{-14}	10^{-14}	10^{-14}	8
BRATU2D	3	22	7	10^{-11}	10^{-10}	-10^{-10}	2.58×10^{4}
BRATU2DT	*ov*						
BRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	23
BROYDN3D	3	7	7	10^{-14}	10^{-14}	-10^{-13}	65
BROYDNBD	4	9	9	10^{-9}	10^{-9}	10^{-10}	83
BT1	>200		401	10^{2}	10^{1}	136.1	678
BT10	>200		401	10^{0}	10^{0}	-0.763	439
BT11	9	18	18	10^{-6}	10^{-13}	0.825	21
BT12	8	16	16	10^{-6}	10^{-12}	6.188	19
BT13	>500		1001	10^{1}	10^{-1}	0.2716×10^{-1}	1043
BT2	*ov*						
BT3	1	2	2	10^{-12}	10^{-13}	4.093	9
BT4	228	456	453	10^{-6}	10^{-11}	-45.51	206
BT5	>300		601	10^{1}	10^{2}	0.946×10^{3}	264
BT6	*ov*						
BT7	*ov*						
BT8	>500		816	10^{-3}	10^{-8}	1.0	2555
BT9	>400		801	10^{0}	10^{-2}	-0.1356	437
BYRDSPHR	127	235	234	10^{-6}	10^{-12}	-4.683	114
CBRATU2D	6	14	13	10^{-14}	10^{-15}	-10^{-15}	1251
CBRATU3D	1	2	2	10^{-14}	10^{-15}	10^{-15}	207
CHANDHEQ	>1000		2001	10^{-4}	10^{-9}	-10^{-8}	6.56×10^{4}
CLUSTER	8	29	17	10^{-6}	10^{-8}	10^{-8}	41
COOLHANS	21	117	42	10^{-6}	10^{-10}	10^{-10}	656
DECONVC	4	8	8	10^{-8}	10^{-14}	10^{-14}	703
DEGENLPB	>100			10^{3}	10^{14}		6×10^{4}
DIXCHLNG	*ov*						
DNIEPER	*ov*						
DRUGDISE	*et*						
DTOC1L	8	9	9	10^{-8}	10^{-14}	0.0307	2.9×10^{4}
DTOC1NA	22	31	31	10^{-6}	10^{-11}	0.03205	9.75×10^{4}
DTOC1NB	*et*						
DTOC1NC	*et*						
DTOC1ND	*et*						
DTOC3	1	2	2	10^{-13}	10^{-13}	-10^{-13}	421
DTOC4	3	7	7	10^{-6}	10^{-7}	10^{-7}	1446
EIGMAXC	*et*						
EIGMINC	*et*						
FCCU	1	2	2	10^{-12}	10^{-13}	11.149	58
GENHS28	1	2	2	10^{-13}	10^{-14}	0.9272	20
GILBERT	20	56	39	10^{-6}	10^{-10}	21.154	2297
GRIDNETA	*et*						
HAGER1	1	2	2	10^{-14}	10^{-14}	-10^{-14}	94
HAGER2	1	2	2	10^{-13}	10^{-14}	10^{-13}	95
HAGER3	1	2	2	10^{-13}	10^{-14}	10^{-14}	95
HAGER4	1	2	2	10^{-13}	10^{-14}	-10^{-14}	94
HATFLDG	*et*						
HEART6	*ov*						
HEART8	*ov*						
HIMMELBE	1	3	3	10^{-15}	10^{-15}	-10^{-15}	9
HONG	*ov*						
HS100LNP	*ov*						
HS104LNP	*ov*						
HS107	*ov*						
HS111	>1000		2001	10^{-5}	10^{-13}	-34.862	9.5×10^{3}
HS111LNP	>1000		2001	10^{-5}	10^{-13}	-34.862	9.5×10^{3}
HS112	8	16	9	10^{-11}	10^{-15}	-47.76	49
HS119	*ov*						
HS26	>300		601	10^{-4}	10^{2}	19.328	227
HS27	19	38	38	10^{-6}	10^{-13}	0.04	21
HS28	1	2	2	10^{-14}	10^{-14}	10^{-14}	8
HS39	>400		801	10^{0}	10^{-2}	-0.136	437

Tabela A.6: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	4	9	9	10^{-6}	10^{-6}	-0.25	13
HS41	167	242	220	10^{-6}	10^{-15}	2.0	562
HS42	4	9	9	10^{-6}	10^{-8}	13.858	12
HS46	>500		858	10^{-4}	10^{-10}	0.9×10^{-6}	1.5×10^{3}
HS47	>500		833	10^{-5}	10^{-11}	0.7×10^{-9}	698
HS48	1	2	2	10^{-14}	10^{-14}	-10^{-15}	8
HS49	16	17	17	10^{-6}	10^{-15}	10^{-8}	20
HS50	9	10	10	10^{-14}	10^{-14}	10^{-15}	16
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	8
HS52	1	2	2	10^{-13}	10^{-14}	5.3266	8
HS53	1	2	2	10^{-13}	10^{-14}	4.093	9
HS54	5	9	9	10^{-9}	10^{-11}	-0.9035	13
HS55	3	22	7	10^{-14}	10^{-15}	6.(6)	183
HS56	>700		1074	10^{-4}	10^{-14}	-0.33×10^{-6}	3253
HS6	*ov*						
HS60	25	42	42	10^{-6}	10^{-13}	0.03257	24
HS61	9	19	18	10^{-7}	10^{-14}	-81.919	20
HS62	9	30	10	10^{-11}	10^{-16}	-0.26×10^{5}	16
HS63	>300		601	10^{1}	10^{2}	0.946×10^{3}	268
HS68	>400		801	10^{0}	10^{-2}	-0.4475	1245
HS69	3	22	7	10^{-6}	0.00	-0.315×10^{-3}	21
HS7	13	22	22	10^{-6}	10^{-13}	-1.732	15
HS77	*ov*						
HS78	6	11	11	10^{-6}	10^{-12}	-2.9197	15
HS79	>500		1001	10^{3}	10^{1}	426.08	2046
HS8	34	79	69	10^{-9}	10^{-9}	-0.(9)	176
HS80	8	17	17	10^{-7}	10^{-9}	0.5395×10^{-1}	20
HS81	9	15	15	10^{-6}	10^{-12}	0.5395×10^{-1}	20
HS9	3	5	5	10^{-10}	0.00	-0.5	9
HS99	>700		761	10^{-7}	10^{-10}	-0.8×10^{9}	7720
HS99EXP	>100						
HYDCAR6	>100						
HYPCIR	4	9	9	10^{-9}	10^{-9}	-10^{-8}	10
INTEGREQ	3	7	7	10^{-7}	10^{-7}	-0.23×10^{-6}	2.68×10^{4}
LCH	1	2	2	10^{-9}	10^{-15}	-3.099	2×10^{4}
LIN	*ov*						
LOTSCHD	*et*						
LSNNODOC	*ov*						
MARATOS	5	9	9	10^{-6}	10^{-12}	-1.0	10
METHANB8	2	4	4	10^{-10}	10^{-10}	-10^{-10}	2125
METHANL8	3	6	6	10^{-11}	10^{-10}	-10^{-11}	3182
MINPERM	*et*						
MSQRTA	>400		801	10^{1}	10^{0}	0.2179×10^{-1}	595
MSQRTB	>900		1801	10^{1}	10^{1}	0.1819×10^{-1}	10^{4}
ODFITS	7	8	8	10^{-8}	10^{-12}	-0.238×10^{4}	47
OPTCNTRL	10	20	20	10^{-6}	10^{-10}	10^{-9}	6089
ORTHREGB	>100						
ORTHREGC	*ov*						
ORTHREGD	*ov*						
ORTHREGE	>100						
PFIT1	*ov*						
PFIT2	*ov*						
PFIT3	*ov*						
PFIT4	*ov*						
PORTFL1	1	2	2	10^{-14}	10^{-15}	0.1869×10^{-1}	14
PORTFL2	1	2	2	10^{-14}	0.00	0.2683×10^{-1}	15
PORTFL3	1	2	2	10^{-15}	10^{-15}	0.2713×10^{-1}	15
PORTFL4	1	2	2	10^{-15}	10^{-16}	0.2123×10^{-1}	15
PORTFL6	1	2	2	10^{-14}	10^{-16}	0.2231×10^{-1}	15
POWELLBS	9	32	18	10^{-8}	10^{-14}	-0.9×10^{-15}	17
POWELLSQ	>200		401	10^{2}	10^{2}	0.1787×10^{2}	152
READING5	1	17	2	10^{-11}	10^{-13}	0.2112×10^{-12}	5.8×10^{4}
RECIPE	21	43	43	10^{-6}	10^{-12}	0.47×10^{-12}	124
RK23	*ov*						
ROBOT	4	8	8	10^{-10}	10^{-10}	-0.4×10^{-10}	37
SEMICON1	1	2	2	10^{-10}	10^{-12}	-0.4×10^{-12}	32
SEMICON2	1	2	2	10^{-12}	10^{-14}	0.36×10^{-14}	33
TENBARS2	*ov*						
TENBARS3	*ov*						
TRIGGER	7	26	15	10^{-14}	10^{-11}	-0.23×10^{-11}	70
ZANGWIL3	1	2	2	0.00	10^{-13}	0.19×10^{-13}	8

Tabela A.7: Tabela com resultados do NPSOL e Curvilínea com PCCAA

Problema	nit	nf	f.o.*	nit	nf	f.o.*
AIRCRFTA	5	7	10^{-28}	2	14	10^{-16}
ALJAZZAF	21	36	75	>100		
ARGTRIG	5	5	10^{-30}	4	8	10^{-11}
ARTIF	I=6			5	11	10^{-8}
AVION2	I=1			*et*		
BDVALUE	4	4	10^{-36}	2	4	10^{-13}
BOOTH	I=1			2	3	0.00
BRATU2D	5	5	10^{-35}	3	22	10^{-10}
BRATU2DT	I=6			3	4	10^{-13}
BRATU3D	5	5	10^{-28}	2	8	10^{-15}
BROYDN3D	6	6	10^{-37}	3	7	10^{-13}
BROYDNBD	8	9	10^{-26}	4	9	10^{-10}
BT1	11	23	-1	>100		
BT10	9	11	-1	5	6	-1.0
BT11	13	18	10^{-1}	7	15	0.825
BT12	9	9	6.19	8	16	6.188
BT13	42	56	10^{-14}	>500		
BT2	25	38	10^{-2}	*ov*		
BT3	5	7	4.09	2	3	4.093
BT4	37	43	3.28	18	19	-3.705
BT5	11	16	$9.6\text{x}10^{2}$	307	614	952.14
BT6	14	20	$2.7\text{x}10^{-1}$	*ov*		
BT7	34	58	$3.06\text{x}10^{2}$	*ov*		
BT8	20	21	1	13	14	1.0
BT9	13	15	-1	74	75	-1.0
BYRDSPHR	I=6			24	50	-4.683
CBRATU2D	5	5	10^{-29}	6	13	10^{-15}
CBRATU3D	5	5	10^{-28}	2	8	10^{-15}
CHANDHEQ	19	21	10^{-11}	7	24	10^{-9}
CLUSTER	I=6			7	8	10^{-8}
COOLHANS	14	15	10^{-23}	6	13	10^{-8}
DECONVC	84	132	10^{-3}	4	8	10^{-14}
DEGENLPB	I=1		$-3\text{x}10^{1}$	*et*		
DIXCHLNG	I=6		$-1\text{x}10^{28}$	*et*		
DNIEPER	6	7	$1.8\text{x}10^{4}$	*et*		
DRUGDISE	I=6		10^{24}	*et*		
DTOC1L	23	23	$7.3\text{x}10^{-2}$	8	9	10^{-15}
DTOC1NA	23	23	$7.5\text{x}10^{-2}$	9	18	10^{-10}
DTOC1NB	25	26	$9.8\text{x}10^{-2}$	*et*		
DTOC1NC	14	15	$3.12\text{x}10^{-1}$	*et*		
DTOC1ND	34	61	$4.14\text{x}10^{-1}$	*et*		
DTOC3	8	9	$2.24\text{x}10^{2}$	2	3	10^{-14}
DTOC4	8	9	3.75	4	8	10^{-14}
EIGMAXC	I=3		$2.3\text{x}10^{14}$	8	9	-1.0
EIGMINC	7	8	1.0	8	9	1.0
FCCU	I=1		0.00	2	7	11.149
GENHS28	I=1		11.6	2	4	0.927
GILBERT	I=6		10^{15}	18	52	21.154
GRIDNETA	19	19	71.70	*et*		
HAGER1	8	10	$8.809\text{x}10^{-1}$	2	3	10^{-15}
HAGER2	7	8	$4.32\text{x}10^{-1}$	2	3	10^{-15}
HAGER3	I=6		1.4714	2	3	10^{-15}
HAGER4	I=1		$4.013\text{x}10^{2}$	2	3	10^{-15}
HATFLDG	I=6		0	814	1628	10^{-14}
HEART6	I=6		10^{18}	*ov*		
HEART8	13	32	0	*ov*		
HIMMELBE	I=1		0	2	12	10^{-15}
HONG	10	18	22.57	16	17	22.571
HS100LNP	I=6		$-2.3\text{x}10^{8}$	*ov*		
HS104LNP	14	19	3.95	Saiu	do	CUTE
HS107	I=6		10^{16}	*ov*		
HS111	41	63	-47.7	998	2095	-34.862
HS111LNP	40	60	-47.7	998	2095	-34.862
HS112	17	36	-47.7	7	15	-47.76
HS119	12	15	244.89	*ov*		
HS26	45	56	$5\text{x}10^{-20}$	947	1910	0.86×10^{-5}
HS27	I=6		1.37x10	19	38	0.04
HS28	5	6	10^{-32}	2	3	10^{-15}
HS39	I=6		$2.4\text{x}10^{5}$	85	86	-1.0

Tabela A.7: Continuação

Problema	nit	nf	f.o.*	nit	nf	f.o.*
HS40	7	9	-0.25	5	10	-0.25
HS41	I=1		2.0	171	246	2.0
HS42	7	10	13.85	5	10	13.858
HS46	I=6		$5.08\text{x}10^{-2}$	377	815	0.84×10^{-5}
HS47	I=6		2.83	40	89	0.296×10^{-7}
HS48	I=7			2	3	10^{-15}
HS49	I=6		10.27	24	25	10^{-15}
HS50	16	19	$8\text{x}10^{-15}$	9	19	10^{-15}
HS51	I=1		-3.0	2	3	10^{-15}
HS52	I=7			2	3	5.327
HS53	I=7			2	3	4.093
HS54	I=1		0	5	9	-0.9035
HS55	I=1		5.638	3	22	6.(6)
HS56	12	18	-3.456	11	30	-0.176×10^{-6}
HS6	I=7			>200		
HS60	I=6		-1.52x10	17	34	0.326×10^{-1}
HS61	10	12	-143.64	9	19	-0.819×10^{2}
HS62	10	16	$-2.6\text{x}10^{4}$	6	8	-0.263×10^{5}
HS63	I=7			307	614	0.952×10^{3}
HS68	I=6		-6.4x10	>1000		
HS69	15	23	$-9.5\text{x}10^{2}$	4	23	-0.315×10^{-3}
HS7	12	15	-1.732	9	18	-0.173×10^{1}
HS77	I=7			>500		
HS78	14	23	-2.92	5	10	-2.919
HS79	13	15	$7.87\text{x}10^{-2}$	>100		
HS8	31	93	-1	8	9	-1.0
HS80	9	10	$5.39\text{x}10^{-2}$	7	15	0.54×10^{-1}
HS81	14	19	$5.39\text{x}10^{-2}$	6	12	0.539×10^{-1}
HS9	6	8	$-5\text{x}10^{-1}$	3	5	-0.5
HS99	I=6		$4.5\text{x}10^{25}$	6	7	-0.831×10^{9}
HS99EXP	I=3		10^{16}	*et*		
HYDCAR6	I=6		10^{32}	*et*		
HYPCIR	7	9	10^{-25}	5	10	10^{-15}
INTEGREQ	4	4	10^{-32}	3	7	-0.23×10^{-6}
LCH	174	395	-4.288	2	3	-3.099
LIN	I=1		$-1.75\text{x}10^{-2}$	*ov*		
LOTSCHD	I=2			*et*		
LSNNODOC	I=1		$1.23\text{x}10^{2}$	*et*		
MARATOS	5	5	-1.0	4	8	-1.0
METHANB8	I=3		10^{29}	2	4	10^{-10}
METHANL8	I=3		10^{32}	3	6	10^{-12}
MINPERM	21	24	$3.84\text{x}10^{-2}$	*et*		
MSQRTA	13	43	10^{-20}	6	7	10^{-13}
MSQRTB	6	8	10^{-29}	*ov*		
ODFITS	I=6		-9.07	7	8	-0.238×10^{4}
OPTCNTRL	5	5	550	15	25	10^{-12}
ORTHREGB	6	6	10^{-19}	*et*		
ORTHREGC	18	19	0.399	*et*		
ORTHREGD	I=6		32.5	*et*		
ORTHREGE	I=4		3.87	*et*		
PFIT1	5 I=6		10^{18}	*et*		
PFIT2	I=6		10^{23}	*et*		
PFIT3	I=6		10^{20}	*et*		
PFIT4	I=6		10^{25}	*et*		
PORTFL1	45	45	$2.04\text{x}10^{-2}$	2	5	0.186×10^{-1}
PORTFL2	38	38	$2.97\text{x}10^{-2}$	2	4	0.268×10^{-1}
PORTFL3	25	26	$3.27\text{x}10^{-2}$	2	3	0.271×10^{-1}
PORTFL4	40	40	$2.63\text{x}10^{-2}$	2	3	0.212×10^{-1}
PORTFL6	43	43	$2.57\text{x}10^{-2}$	2	10	0.223×10^{-1}
POWELLBS	I=1		10^{-22}	9	10	10^{-12}
POWELLSQ	I=7			>200		
READING5	7	8	0^{-29}	5	80	10^{-14}
RECIPE	I=6			11	12	10^{-8}
RK23	7	18	$8.3\text{x}10^{-2}$	*et*		
ROBOT	15	22	5.462	5	9	10^{-15}
SEMICON1	62	139	10^{-18}	2	3	10^{-13}
SEMICON2	I=3		10^{21}	2	3	0.00
TENBARS2	I=6		10^{7}	*et*		
TENBARS3	I=6		10^{27}	*et*		
TRIGGER	21	37	10^{-17}	6	14	10^{-11}
ZANGWIL3	I=1			2	3	10^{-30}

Tabela A.8: Tabela com resultados do MATLAB e Curvilínea com PCCAA

Problema	nf	f.o.*	$it > 100n$	nf	f.o.*
AIRCRFTA	4	0		14	10^{-16}
ALJAZZAF	>293	53.94	>100	>100	
ARGTRIG	6	0		9	0.00
ARTIF	11	0		12	10^{-15}
AVION2	*ov*			*et*	
BDVALUE	4	0		5	10^{-13}
BOOTH	3	0		3	0.00
BRATU2D	5	0		23	10^{-10}
BRATU2DT	5	0		7	10^{-11}
BRATU3D	5	0		8	10^{-15}
BROYDN3D	6	0		8	10^{-15}
BROYDNBD	8	0		10	10^{-14}
BT1	>193	-0.986	>200	>200	
BT10	8	-1		6	-1.0
BT11	17	0.8249		19	0.825
BT12	9	6.188		20	6.180
BT13	236	-10^{10}		>500	
BT2	18	2.19		*ov*	
BT3	11	4.0930		3	4.093
BT4	14	-45.51		28	-3.704
BT5	10	961.7		>300	
BT6	38	0.277		*ov*	
BT7	155	360.38		*ov*	
BT8	19	1		14	1.0
BT9	>399	-5.34	>400	76	-1.0
BYRDSPHR	226	-4.683		33	-4.683
CBRATU2D	5	0		29	10^{-10}
CBRATU3D	5	0		8	10^{-15}
CHANDHEQ	>983	0	>1000	>1000	
CLUSTER	11	0		9	10^{-14}
COOLHANS	>899	0	>900	95	10^{-10}
DECONVC	>3745		*et*	>100	
DEGENLPB	*nc*			*et*	
DIXCHLNG	208	2471.9		*et*	
DNIEPER	>139	10^{15}		*et*	
DRUGDISE	>6299	-1.2947	>6300	*et*	
DTOC1L	42	0.03072		26	10^{-14}
DTOC1NA	42	0.0320		81	10^{-9}
DTOC1NB	43	0.05245		*et*	
DTOC1NC	28	0.2703		*et*	
DTOC1ND	4719	0.3845		*et*	
DTOC3	17	10^{-16}		3	10^{-14}
DTOC4	12	10^{-16}		9	10^{-14}
EIGMAXC	9	-1		9	-1.0
EIGMINC	9	1		9	1.0
FCCU	21	11.1491		7	11.149
GENHS28	10	0.927		4	0.927
GILBERT	65	25.15		65	21.153
GRIDNETA	28	37.99		*et*	
HAGER1	20	10^{-14}		3	10^{-14}
HAGER2	25	10^{-16}		3	10^{-14}
HAGER3	22	10^{-14}		3	10^{-14}
HAGER4	31	10^{-14}		3	10^{-14}
HATFLDG	19	0		1628	10^{-13}
HEART6	*nhsa*			*ov*	
HEART8	*nhsa*			*ov*	
HIMMELBE	4	0		12	10^{-15}
HONG	*nc*			18	22.571
HS100LNP	>699	680.6	>700	*ov*	
HS104LNP	Saiu do CUTE				
HS107	*nc*	10^{29}		>900	
HS111	*nc*	10^{86}		>1000	
HS111LNP	*nc*	10^{88}		>1000	
HS112	*ov*			16	-47.76
HS119	*et*	10^{17}		*ov*	
HS26	36	10^{16}		>1000	
HS27	>295	2.22	>300	48	0.04
HS28	11	10^{-20}		3	10^{-15}
HS39	>399	-5.344	>400	87	-1.0

Tabela A.8: Continuação

Problema	nf	f.o.*	$it > 100n$	nf	f.o.*
HS40	11	-0.25		11	-0.25
HS41	*nc*	10^{59}		408	2.0
HS42	11	13.858		10	13.858
HS46	48	10^{-17}		>500	
HS47	77	10^{-16}		>1000	
HS48	12	10^{-21}		3	10^{-14}
HS49	52	10^{-22}		>200	ver
HS50	22	10^{-20}		12	10^{-15}
HS51	9	10^{-24}		3	10^{-14}
HS52	8	5.3266		3	5.327
HS53	10	4.093		3	4.093
HS54	3	10^{-34}		33	-0.9035
HS55	11	6.132		23	6.(6)
HS56	15	-3.456		>800	
HS6	12	10^{-20}		>200	
HS60	12	0.032		44	0.326×10^{-1}
HS61	*nhsa*			29	-0.819×10^{2}
HS62	*ov*			12	-0.26×10^{5}
HS63	10	961.715		>300	
HS68	362	-184.119		>400	
HS69	*ov*			23	-0.3×10^{-3}
HS7	15	-1.73		23	-0.17×10^{1}
HS77	27	0.2415		>500	
HS78	13	-2.9197		12	-2.919
HS79	22	0.0787		>100	
HS8	7	-1		9	-1.0
HS80	9	0.0539		18	0.5×10^{-1}
HS81	13	0.0539		33	0.5×10^{-1}
HS9	16	-0.5		7	-0.5
HS99	>699	10^{8}	>700	15	-0.831×10^{9}
HS99EXP	>4	10^{16}		*et*	
HYDCAR6	7	0		*et*	
HYPCIR	7	0		10	10^{-15}
INTEGREQ	4	0		294	10^{-15}
LCH	*nc*	10^{39}		3	-3.099
LIN	*ov*			*et*	
LOTSCHD	*nhsa*			*et*	
LSNNODOC	27	18.97		*et*	
MARATOS	5	-1		12	-1.0
METHANB8	5	0		5	10^{-11}
METHANL8	6	0		7	10^{-13}
MINPERM	5	0.0384		*et*	
MSQRTA	8	0		8	
MSQRTB	9	0		*ov*	10^{-15}
ODFITS	36	-2380.03		*et*	
OPTCNTRL	93	10^{-13}		*et*	
ORTHREGB	9	10^{-15}		*et*	
ORTHREGC	21	0.399		*et*	
ORTHREGD	>1132	8142.7		*et*	
ORTHREGE	>3599	4.3455	>3600	*et*	
PFIT1	*nhsa*			*et*	
PFIT2	*nhsa*			*et*	
PFIT3	*nhsa*			*et*	
PFIT4	*nhsa*			*et*	
PORTFL1	65	0.0186		5	0.0187
PORTFL2	64	0.0268		4	0.0268
PORTFL3	65	0.0271		3	0.0271
PORTFL4	65	0.0212		3	0.0212
PORTFL6	65	0.0223		10	0.0223
POWELLBS	14	0		11	10^{-15}
POWELLSQ	>191	0	>200	*et*	
READING5	2	10^{-34}		80	10^{-14}
RECIPE	26	0		20	10^{-14}
RK23	*nhsa*			*et*	
ROBOT	*nhsa*			9	10^{-15}
SEMICON1	>1199	0	>1200	3	10^{-14}
SEMICON2	9	0		3	0.00
TENBARS2	*nhsa*			*et*	
TENBARS3	*nhsa*			*et*	
TRIGGER	9	0		14	10^{-10}
ZANGWIL3	3	0		3	10^{-29}

Tabela A.9: Resultados da versão Linear/Truncada com factor (7.7) e (7.6)

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo	nit	nf	tempo
AIRCRFTA	24	34	25	10^{-7}	10^{-8}	10^{-8}	84	5	6	
ALJAZZAF	>300			10^{3}	10^{0}	50.272	452	>100		549
ARGTRIG	4	5	5	10^{-6}	10^{-7}	0.11×10^{-6}	53	4	5	93
ARTIF	13	39	14	10^{-6}	10^{-6}	0.2×10^{-6}	436	10	30	433
AVION2	*et*							*et*		
BDVALUE	2	3	3	10^{-11}	10^{-10}	10^{-10}	45	2	3	62
BOOTH	1	2	2	10^{-14}	10^{-15}	10^{-15}	5	1	2	12
BRATU2D	2	3	3	10^{-7}	10^{-6}	-0.4×10^{-6}	3825	2	3	3822
BRATU2DT	4	5	5	10^{-7}	10^{-7}	-0.26×10^{-6}	6385	2	3	3821
BRATU3D	3	4	4	10^{-8}	10^{-8}	-10^{-8}	34	3	4	59
BROYDN3D	4	5	5	10^{-8}	10^{-8}	-0.18×10^{-8}	51	4	5	91
BROYDNBD	6	7	5	10^{-6}	10^{-7}	0.69×10^{-7}	72	6	7	136
BT1	>200			10^{4}	10^{3}	10^{5}	175	>100		868
BT10	6	8	7	10^{-7}	10^{-8}	-1.0	8	6	7	60
BT11	741	1426	742	10^{-6}	10^{-7}	0.8248916	915	25	154	323
BT12	24	27	25	10^{-6}	10^{-8}	6.188118	26	14	15	138
BT13	>500			10^{0}	10^{0}	10^{8}	1346	>100		773
BT2	40	41	41	10^{-6}	10^{-13}	0.0325682	22	35	36	342
BT3	1	2	2	10^{-9}	10^{-9}	4.0930	7	2	3	25
BT4	>500			10^{-3}	10^{-8}	-45.51055		>100		522
BT5	17	19	18	10^{-6}	10^{-11}	961.715	14	11	13	113
BT6	133	183	134	10^{-6}	10^{-7}	0.277044	112	>100		498
BT7	>500			10^{13}	10^{4}	10^{10}	7853	>100		2361
BT8	17	64	18	10^{-6}	10^{-6}	1.0	28	14	57	167
BT9	81	156	82	10^{-8}	10^{-10}	-1.0	64	93	186	534
BYRDSPHR	129	358	130	10^{-6}	10^{-13}	-4.683300	107	>100		597
CBRATU2D	3	4	4	10^{-8}	10^{-7}	-10^{-8}	402	3	4	423
CBRATU3D	4	5	5	10^{-8}	10^{-7}	10^{-7}	486	3	4	414
CHANDHEQ	>1000			10^{-4}	10^{-9}	10^{-9}	2×10^{4}	15	27	400
CLUSTER	210	1385	211	10^{-7}	10^{-7}	-10^{-7}	299	43	220	419
COOLHANS	>500			10^{4}	10^{0}	0.18781	9×10^{3}	>100		6355
DECONVC	>300			10^{-3}	10^{-7}	0.2497×10^{-4}	3.6×10^{4}	>100		24084
DEGENLPB	>100					-10^{12}		>100		2.2×10^{5}
DIXCHLNG	>100					10^{20}		>100		7205
DNIEPER	>10			10^{4}	10^{3}	-10^{7}	10^{5}	*et*		
DRUGDISE	*et*							*et*		
DTOC1L	12	33	13	10^{-6}	10^{-6}	0.030726	105	8	11	3.6×10^{4}
DTOC1NA	15	34	16	10^{-3}	10^{-3}	0.032577	105	9	12	3.8×10^{4}
DTOC1NB	*et*							*et*		
DTOC1NC	>50			10^{-2}	10^{-4}	0.2702118	4.3×10^{5}	*et*		
DTOC1ND	>50			10^{-2}	10^{-2}	0.387927	3.5×10^{5}	*et*		
DTOC3	1	2	2	10^{-9}	10^{-9}	10^{-9}	436	2	3	673
DTOC4	4	5	5	10^{-7}	10^{-7}	0.5×10^{-7}	10^{3}	3	4	911
EIGMAXC	10	19	11	10^{-7}	10^{-8}	10^{-8}	2720	43	338	4.4×10^{4}
EIGMINC	99	846	100	10^{-6}	10^{-7}	10.74	10^{5}	86	599	7.9×10^{4}
FCCU	1	2	2	10^{-10}	10^{-10}	11.149	59	2	3	106
GENHS28	1	2	2	10^{-10}	10^{-10}	0.92717	18	2	3	43
GILBERT	>500			10^{-3}	10^{-3}	21.1531	4×10^{4}	53	131	6236
GRIDNETA	10	41	11	10^{2}	10^{1}	0.309565×10^{3}	2.65×10^{5}	*et*		
HAGER1	1	2	2	10^{-9}	10^{-10}	10^{-11}	96	2	3	162
HAGER2	1	2	2	10^{-8}	10^{-10}	10^{-11}	96	2	3	163
HAGER3	1	2	2	10^{-7}	10^{-9}	10^{-10}	96	2	3	162
HAGER4	1	2	2	10^{-7}	10^{-9}	10^{-10}	97	2	3	162
HATFLDG	6	20	7	10^{-14}	10^{-7}	10^{-12}	4450	6	8	2026
HEART6	>100			10^{20}	10^{8}	-10^{7}	10^{3}	>100		1219
HEART8	>1000			10^{4}	10^{1}	6.161	9×10^{3}	>100		1883
HIMMELBE	2	3	3	10^{-15}	10^{-14}	10^{-15}	6	2	3	23
HONG	*et*							*et*		
HS100LNP	>1000			10^{-4}	10^{-6}	680.63	1.4×10^{3}	77	127	524
HS104LNP	*ov*							bloq		
HS107	*et*					10^{34}		>100		3746
HS111	>1000			10^{-2}	10^{-4}	-47.76	1.5×10^{4}	>100		1306
HS111LNP	>1000			10^{-2}	10^{-4}	-47.76		>100		1162
HS112	*ov*							bloq.		
HS119	*ov*							bloq.		
HS26	15638		15639	10^{-13}	10^{-6}	10^{-8}	6×10^{3}	>100		564
HS27	*nc*					10^{18}		>100		570
HS28	1	2	2	10^{-13}	10^{-14}	10^{-14}	52	1	2	15
HS39	81	157	82	10^{-9}	10^{-12}	-1.0	796	93	186	513

Tabela A.9: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo	nit	nf	tempo
HS40	14	16	15	10^{-6}	10^{-7}	-0.24(9)	357	5	6	52
HS41	>100					10^{20}		>100		512
HS42	56	72	57	10^{-6}	10^{-13}	13.858	696	39	55	383
HS46	>2000			10^{-4}	10^{-5}	10^{-5}	1.3×10^{4}	>2000		
HS47	>100					10^{11}		>100		857
HS48	1	2	2	10^{-14}	10^{-13}	10^{-15}	66	2	3	26
HS49	16	17	17	10^{-6}	0.00	10^{-8}	340	12	13	137
HS50	9	10	10	10^{-14}	10^{-14}	10^{-15}	329	8	9	86
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	69	1	2	16
HS52	>100					10^{73}		>100		548
HS53	1	2	2	10^{-9}	10^{-9}	4.093	69	2	3	24
HS54	>1000			10^{-3}	10^{-11}		7×10^{3}	7	26	74
HS55	2	3	3	10^{-10}	10^{-10}	6.(6)	187	2	3	43
HS56	19	23	20	10^{-6}	10^{-12}	-3.456	440	14	18	174
HS6	239	284	240	10^{-6}	10^{-9}	-10^{-9}	1353	>100		521
HS60	35	36	36	10^{-6}	10^{-13}	0.032568	466	22	23	216
HS61	19	43	20	10^{-6}	10^{-13}	-143.646	431	15	39	157
HS62	>10000					10^{5}		>100		721
HS63	17	19	18	10^{-6}	10^{-11}	961.715	334	11	13	111
HS68	*ov*							26	38	250
HS69	*ov*							*bloq*		
HS7	25	26	26	10^{-6}	10^{-12}	-1.73205	427	19	20	181
HS77	461	584	462	10^{-6}	10^{-7}	0.2415	3.28×10^{3}	>100		642
HS78	8	9	9	10^{-6}	10^{-11}	-2.9197	345	6	7	67
HS79	>1000			10^{-2}	10^{-4}	0.07878	7.7×10^{3}	19	37	205
HS8	4	6	5	10^{-8}	10^{-7}	-0.(9)	183	4	6	42
HS80	113	114	114	10^{-6}	10^{-12}	0.05395	1165	60	61	611
HS81	105	106	106	10^{-6}	10^{-12}	0.05395	1091	59	60	546
HS9	4	5	5	10^{-10}	0.00	-0.5	171	3	4	36
HS99	>1000			10^{0}	10^{-4}	-10^{-9}	8.7×10^{3}	>100		563
HS99EXP	>10			10^{7}	10^{7}	-10^{-14}	2×10^{4}	>100		2.7×10^{5}
HYDCAR6	>100			10^{10}	10^{4}	-0.4×10^{4}	4.9×10^{5}	>100		3.4×10^{5}
HYPCIR	4	6	5	-10^{-9}	-10^{-8}	10^{-9}	7	4	6	41
INTEGREQ	2	3	3	10^{-6}	10^{-6}	0.49×10^{-5}	10^{4}	13	71	2.7×10^{5}
LCH	*et*							>100		3.98×10^{6}
LIN	*et*							*bloq*		
LOTSCHD	>100			10^{5}	10^{8}	10^{13}	3.4×10^{4}	>100		8341
LSNNODOC	*et*							>100		1074
MARATOS	17	18	18	10^{-6}	10^{-12}	-1.0	10	11	12	105
METHANB8	5	9	6	10^{-4}	10^{-5}	-0.67×10^{-5}	4.9×10^{3}	47	118	6.17×10^{4}
METHANL8	5	6	6	10^{-7}	10^{-9}	0.9×10^{-9}	3.5×10^{3}	77	245	1.26×10^{5}
MINPERM	>10			10^{-4}	10^{-4}	0.0384	10^{5}	3	4	1.768×10^{4}
MSQRTA	>400			10^{0}	10^{-1}	0.0682477	10^{3}	>100		806
MSQRTB	201	1533	202	10^{-7}	10^{-7}	0.286×10^{-7}	7×10^{3}	>100		4846
ODFITS	7	8	8	10^{-8}	10^{-12}	-0.238×10^{4}	44	6	7	99
OPTCNTRL	43	74	44	10^{-6}	10^{-9}	10^{-8}	2×10^{4}	5	6	1988
ORTHREGB	30	46	31	10^{-7}	10^{-8}	10^{-9}	2×10^{3}	15	37	1687
ORTHREGC	>1000			10^{-1}	10^{-3}	2.2319	1.8×10^{5}	>100		2×10^{4}
ORTHREGD	213	369	214	10^{-6}	10^{-10}	3.412121	2×10^{4}	>100		7998
ORTHREGE	>100			10^{-1}	10^{-2}	14.869	9×10^{4}	>100		2.23×10^{5}
PFIT1	*et*							>100		524
PFIT2	>500			10^{25}	10^{1}	-2.765	631	*ov*		
PFIT3	>500			10^{4}	10^{2}	9.777	834	>100		712
PFIT4	>500			10^{5}	10^{2}	18.28	697	>100		586
PORTFL1	1	2	2	10^{-12}	10^{-16}	0.186865×10^{-1}	13	2	3	35
PORTFL2	1	2	2	10^{-12}	0.00	0.26828×10^{-1}	13	2	3	36
PORTFL3	1	2	2	10^{-12}	10^{-15}	0.2713×10^{-1}	12	2	3	35
PORTFL4	1	2	2	10^{-12}	10^{-16}	0.2123×10^{-1}	13	2	3	35
PORTFL6	1	2	2	10^{-13}	10^{-15}	0.223×10^{-1}	13	2	3	36
POWELLBS	19	31	20	10^{-8}	10^{-13}	10^{-13}	14	31	179	327
POWELLSQ	>1000			10^{1}	10^{-3}	0.574×10^{-1}	485	>100		487
READING5	1	3	2	10^{-11}	10^{-13}	10^{-13}	10^{4}	1	3	1.09×10^{4}
RECIPE	>1000			10^{0}	10^{-1}	0.142068×10^{-1}	2×10^{3}	>100		803
RK23	>100			10^{28}	10^{17}	10^{17}	2.2×10^{4}	>100		1.6×10^{4}
ROBOT	36	242	37	10^{-6}	10^{-7}	-10^{-7}	535	44	259	987
SEMICON1	2	3	3	10^{-11}	10^{-13}	10^{-13}	44	2	3	63
SEMICON2	1	2	2	10^{-12}	10^{-13}	10^{-13}	31	2	3	63
TENBARS2	>100			10^{1}	10^{1}	-10^{5}	1.9×10^{4}	>100		7161
TENBARS3	>100			10^{1}	10^{1}	-10^{5}	1.9×10^{4}	>100		7158
TRIGGER	45	106	46	10^{-3}	10^{-7}	10^{-8}	215	7	8	94
ZANGWIL3	1	2	2	0.00	10^{-12}	10^{-13}	5	1	2	13

Tabela A.10: Resultados da versão Curvilínea/Truncada com factor (7.7)

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
AIRCRFTA	2	5	5	10^{-8}	10^{-14}	10^{-14}	12
ALJAZZAF	>1000		2001	10^{3}	10^{0}	51.7	1499
ARGTRIG	2	5	5	10^{-6}	10^{-7}	10^{-6}	24
ARTIF	6	8	13	10^{-7}	10^{-8}	10^{-8}	59
AVION2	>100			10^{7}	10^{0}	10^{8}	10^{4}
BDVALUE	1	3	3	10^{-10}	10^{-10}	10^{-10}	15
BOOTH	1	2	2	0.00	10^{-14}	10^{-14}	8
BRATU2D	1	3	3	10^{-7}	10^{-6}	10^{-6}	128
BRATU2DT	2	5	5	10^{-12}	10^{-15}	10^{-15}	21
BRATU3D	2	5	5	10^{-15}	10^{-15}	10^{-15}	15
BROYDN3D	2	5	5	10^{-9}	10^{-9}	10^{-9}	23
BROYDNBD	4	9	9	10^{-8}	10^{-10}	10^{-10}	38
BT1	>200		401	10^{0}	10^{-9}	0.09733	537
BT10	5	25	10	10^{-12}	10^{-12}	-1.0	21
BT11	>100		201	10^{-3}	10^{-4}	0.824958	507
BT12	5	11	11	10^{-6}	10^{-8}	6.188188	22
BT13	>500		998	10^{0}	10^{-3}	-10^{-6}	2577
BT2	33	56	56	10^{-6}	10^{-13}	0.032568	108
BT3	1	3	3	10^{-13}	0.00	4.0930	8
BT4	103	206	206	10^{-9}	10^{-10}	-45.51055	336
BT5	9	17	17	10^{-6}	10^{-11}	961.71517	33
BT6	>500		1001	10^{1}	10^{-6}	0.629738	10^{3}
BT7	*ov*						
BT8	>500		1001	10^{0}	10^{-2}	1.002783	1374
BT9	10	22	21	10^{-6}	10^{-7}	-1.0	37
BYRDSPHR	77	166	151	10^{-7}	10^{-8}	-4.683300	252
CBRATU2D	2	5	5	10^{-13}	10^{-12}	10^{-12}	28
CBRATU3D	2	5	5	10^{-15}	10^{-15}	10^{-15}	30
CHANDHEQ	>1000		2001	10^{-2}	10^{-4}	10^{-4}	1.3×10^{4}
CLUSTER	7	40	15	10^{-9}	10^{-11}	10^{-11}	30
COOLHANS	>20		41	10^{5}	10^{1}	-3.9453	201
DECONVC	>1000		1882	10^{-5}	10^{-11}	10^{-8}	5.5×10^{4}
DEGENLPB	>1000		2001	10^{3}	10^{-7}	-10^{-11}	10^{4}
DIXCHLNG	*et*						
DNIEPER	>1000		2001	10^{4}	10^{-8}	-10^{-6}	1.6×10^{5}
DRUGDISE	>100		201	10^{6}	10^{0}	0.4760	1.5×10^{6}
DTOC1L	>100		201	10^{-4}	10^{-4}	0.030599	4×10^{4}
DTOC1NA	29	153	59	10^{-5}	10^{-6}	0.0320538	1.2×10^{4}
DTOC1NB	>100		201	10^{0}	10^{-2}	0.13399	4×10^{4}
DTOC1NC	60	164	121	10^{-6}	10^{-10}	0.270314	2.2×10^{4}
DTOC1ND	80	207	161	10^{-6}	10^{-15}	0.384522	3.2×10^{4}
DTOC3	1	3	3	10^{-9}	10^{-15}	10^{-15}	76
DTOC4	3	7	7	10^{-6}	10^{-9}	10^{-10}	171
EIGMAXC	4	9	9	10^{-9}	10^{-15}	10^{-15}	237
EIGMINC	4	9	9	10^{-10}	10^{-14}	10^{-13}	240
FCCU	1	2	2	10^{-10}	10^{-10}	11.149109	15
GENHS28	1	2	2	10^{-9}	10^{-10}	0.9271736	12
GILBERT	22	100	45	10^{-6}	10^{-7}	21.1535	289
GRIDNETA	1	3	3	10^{-13}	10^{-14}	37.997	405
HAGER1	1	3	3	10^{-9}	10^{-15}	10^{-15}	20
HAGER2	1	2	2	10^{-8}	10^{-10}	10^{-12}	22
HAGER3	1	3	3	10^{-7}	10^{-15}	10^{-13}	25
HAGER4	1	3	3	10^{-7}	10^{-15}	10^{-13}	25
HATFLDG	88	176	176	10^{-10}	10^{-10}	10^{-10}	2373
HEART6	>500		1001	10^{6}	10^{1}	6.13589	4718
HEART8	>500		1001	10^{6}	10^{5}	-0.33×10^{4}	3290
HIMMELBE	1	3	3	10^{-15}	10^{-14}	10^{-15}	8
HONG	*et*						
HS100LNP	386	813	772	10^{-6}	10^{-9}	680.6300	909
HS104LNP	Saiu	do	CUTE				
HS107	>100		201	10^{11}	10^{6}	10^{14}	580
HS111	>500		858	10^{-4}	10^{-10}	-47.761	2.9×10^{3}
HS111LNP	>500		858	10^{-4}	10^{-10}	-47.761	2.7×10^{3}
HS112	*et*						
HS119	*et*						
HS26	>1000		1970	10^{-4}	10^{-10}	10^{-7}	1344
HS27	14	28	28	10^{-6}	10^{-13}	0.04	49
HS28	1	2	2	10^{-13}	10^{-13}	10^{-13}	7
HS39	10	22	21	10^{-6}	10^{-7}	-1.0	37

Tabela A.10: Continuação

Problema	nit	nf	ng	$\|\nabla L\|_2$	$\|c\|_2$	f.o.*	tempo
HS40	5	11	11	10^{-8}	10^{-9}	-0.25	21
HS41	*et*						
HS42	11	49	22	10^{-6}	10^{-13}	13.8578	44
HS46	>1000		1979	10^{-4}	10^{-8}	10^{-6}	1.8×10^{3}
HS47	772	1539	1534	10^{-6}	10^{-9}	10^{-9}	1.7×10^{3}
HS48	1	2	2	10^{-14}	10^{-14}	10^{-14}	7
HS49	14	17	17	10^{-6}	10^{-15}	10^{-8}	49
HS50	7	11	11	10^{-13}	10^{-14}	10^{-15}	29
HS51	1	2	2	10^{-14}	10^{-14}	10^{-14}	7
HS52	2	4	4	10^{-12}	10^{-13}	5.3266	11
HS53	1	2	2	10^{-11}	10^{-11}	4.093	7
HS54	>1000		1998	10^{-3}	10^{-8}	-0.63769	1.7×10^{3}
HS55	6	8	8	10^{-9}	10^{-11}	6.13244	25
HS56	6	12	12	10^{-6}	10^{-12}	-3.456	27
HS6	>500		1001	10^{1}	10^{-4}	0.8266	897
HS60	32	52	52	10^{-6}	10^{-13}	0.032568	104
HS61	8	18	17	10^{-6}	10^{-8}	-143.646	31
HS62	8	14	11	10^{-8}	10^{-14}	-0.262725	29
HS63	9	17	17	10^{-6}	10^{-11}	961.715	33
HS68	12	56	25	10^{-9}	10^{-14}	0.119×10^{-5}	52
HS69	6	47	13	10^{-8}	10^{-12}	-0.39×10^{-4}	31
HS7	10	19	19	10^{-6}	10^{-12}	-1.73205	35
HS77	46	110	92	10^{-6}	10^{-10}	0.24150	165
HS78	5	9	9	10^{-6}	10^{-12}	-2.9197	21
HS79	50	216	101	10^{-5}	10^{-6}	0.0787769	202
HS8	3	6	6	10^{-12}	10^{-12}	-1.0	15
HS80	7	15	15	10^{-7}	10^{-9}	0.0539498	30
HS81	5	11	11	10^{-6}	10^{-8}	0.0539498	22
HS9	4	5	5	10^{-10}	0.00	-0.5	16
HS99	>500		1001	10^{-3}	10^{-10}	-0.831079×10^{9}	4487
HS99EXP	>100			10^{15}	10^{14}	10^{28}	3.4×10^{4}
HYDCAR6	>20		41	10^{1}	10^{0}	0.353837	3×10^{4}
HYPCIR	3	6	6	10^{-11}	10^{-11}	10^{-11}	14
INTEGREQ	1	3	3	10^{-6}	10^{-6}	0.488×10^{-5}	364
LCH	>500		860	10^{-4}	10^{-13}	-4.2887	3.8×10^{5}
LIN	*et*						
LOTSCHD	>100		201	10^{6}	10^{7}	10^{11}	588
LSNNODOC	>25		46	10^{-1}	10^{-3}	18.985	91
MARATOS	6	10	10	10^{-6}	10^{-12}	-1.0	23
METHANB8	>100		201	10^{2}	10^{2}	-0.12×10^{2}	6×10^{4}
METHANL8	>20		41	10^{5}	10^{3}	0.275×10^{3}	4.5×10^{4}
MINPERM	1	3	3	10^{-11}	10^{-13}	0.0384	200
MSQRTA	277	666	555	10^{-8}	10^{-10}	10^{-10}	652
MSQRTB	>400		801	10^{2}	10^{1}	4.5797	3×10^{3}
ODFITS	7	9	9	10^{-8}	10^{-12}	-2380.0	35
OPTCNTRL	153	743	307	10^{-6}	10^{-11}	10^{-8}	8×10^{3}
ORTHREGB	5	11	11	10^{-6}	10^{-15}	10^{-14}	63
ORTHREGC	70	144	141	10^{-6}	10^{-15}	0.3990	10^{3}
ORTHREGD	>500		1001	10^{1}	10^{-2}	27.94579	10^{4}
ORTHREGE	>1000		2001	10^{1}	10^{-3}	5.24289	6×10^{4}
PFIT1	7	15	15	10^{-8}	10^{-8}	10^{-8}	29
PFIT2	>300		601	10^{5}	10^{2}	-0.7145×10^{2}	629
PFIT3	>300		601	10^{2}	10^{1}	3.1354	649
PFIT4	>300		601	10^{5}	10^{2}	87.41611	726
PORTFL1	1	2	2	10^{-13}	10^{-15}	0.0186865	14
PORTFL2	1	2	2	10^{-12}	10^{-15}	0.026828	14
PORTFL3	1	2	2	10^{-12}	10^{-15}	0.0271304	13
PORTFL4	1	2	2	10^{-12}	10^{-15}	0.0212309	14
PORTFL6	1	2	2	10^{-12}	10^{-16}	0.0223088	14
POWELLBS	7	60	15	10^{-6}	10^{-3}	-0.1012×10^{-3}	33
POWELLSQ	>300		601	10^{2}	10^{1}	9.2374	587
READING5	1	3	3	10^{-6}	10^{-17}	10^{-17}	2×10^{3}
RECIPE	>1000		2001	10^{-2}	10^{-4}	0.9×10^{-4}	1.8×10^{3}
RK23	>300		601	10^{2}	10^{1}	-0.221×10^{3}	3×10^{3}
ROBOT	5	11	11	10^{-9}	10^{-10}	10^{-9}	26
SEMICON1	1	3	3	10^{-6}	10^{-13}	10^{-13}	13
SEMICON2	1	2	2	10^{-12}	10^{-13}	10^{-13}	12
TENBARS2	>500		1001	10^{1}	10^{-4}	-10^{-6}	4×10^{3}
TENBARS3	>500		1001	10^{1}	10^{-4}	-10^{-6}	4×10^{3}
TRIGGER	>300		601	10^{2}	10^{-1}	-0.77×10^{-1}	1.7×10^{3}
ZANGWIL3	1	2	2	0.00	10^{-14}	10^{-14}	8

Tabela A.11: Resultados da versão Curvilínea/Truncada com factor (7.6)

Problema	nit	nf	$\|\nabla L\|_2$	tempo
AIRCRFTA	>100		10^{2}	
ALJAZZAF	>100		10^{3}	
ARGTRIG	2	5	10^{-6}	179
ARTIF	4	9	10^{-9}	374
AVION2	>100		10^{7}	
BDVALUE	3	18	10^{-10}	290
BOOTH	2	4	10^{-14}	78
BRATU2D	3	21	10^{-9}	828
BRATU2DT	3	22	10^{-5}	837
BRATU3D	2	5	10^{-14}	261
BROYDN3D	3	7	10^{-13}	262
BROYDNBD	>100		10	2.4×10^{3}
BT1	16	36	10^{-7}	383
BT10	5	25	10^{-6}	283
BT11	10	22	10^{-4}	394
BT12	>100		10^{2}	1.1×10^{3}
BT13	>100		10^{7}	1.2×10^{3}
BT2	33	67	10^{-4}	592
BT3	3	6	10^{-13}	221
BT4	35	82	10^{-7}	497
BT5	7	15	10^{-4}	227
BT6	>100		10^{2}	1201
BT7	>100		10^{3}	1392
BT8	>100		10	1223
BT9	>100		10^{-1}	915
BYRDSPHR	>100		10^{1}	1156
CBRATU2D	1	3	10^{-4}	239
CBRATU3D	2	5	10^{-14}	414
CHANDHEQ	>400		10^{-3}	9021
CLUSTER	4	21	10^{-4}	228
COOLHANS	>100		10^{2}	2×10^{3}
DECONVC	>100		10^{-1}	10^{4}
DEGENLPB	>100		10^{6}	
DIXCHLNG	28	102	10^{-4}	771
DNIEPER	>100	10^{9}		
DRUGDISE	>100	10^{6}		10^{5}
DTOC1L	>100	10^{-1}		4×10^{4}
DTOC1NA	>100	10^{-2}		4.5×10^{4}
DTOC1NB	>100	10^{93}		8.7×10^{4}
DTOC1NC	>100	10^{-1}		4×10^{4}
DTOC1ND	>100		10^{1}	4×10^{4}
DTOC3	>100		10^{-2}	10^{4}
DTOC4	>100		10^{-2}	10^{4}
EIGMAXC	>100		10^{-2}	9.5×10^{3}
EIGMINC	15	45	10^{-5}	1.7×10^{3}
FCCU	>100		10^{-1}	3×10^{3}
GENHS28	3	7	10^{-7}	352
GILBERT	16	42	10^{-4}	10^{3}
GRIDNETA	>100		10^{-2}	4×10^{4}
HAGER1	3	7	10^{-4}	361
HAGER2	3	7	10^{-5}	340
HAGER3	3	7	10^{-6}	348
HAGER4	3	7	10^{-5}	432
HATFLDG	>100		10^{-1}	6.8×10^{3}
HEART6	>100		10^{4}	1.8×10^{3}
HEART8	>100		10^{2}	1.9×10^{3}
HIMMELBE	3	7	10^{-15}	198
HONG	6	22	10^{-6}	358
HS100LNP	>100		10^{2}	1.5×10^{3}
HS104LNP	>100		10^{-3}	1.6×10^{3}
HS107	>100		10^{13}	2.9×10^{3}
HS111	>100		10^{1}	2.7×10^{3}
HS111LNP	>100		10^{1}	2.7×10^{3}
HS112	*ov*			
HS119	10	60	10^{-4}	656
HS26	142	284	10^{-4}	1266
HS27	5	11	10^{-5}	292
HS28	2	3	10^{-13}	109
HS39	>100		10^{-1}	1073

Tabela A.11: Continuação

Problema	nit	nf	$\|\nabla L\|_2$	tempo
HS40	4	21	10^{-4}	279
HS41	>100		10^{0}	940
HS42	5	25	10^{-6}	333
HS46	>100		10^{-1}	1068
HS47	83	185	10^{-4}	1021
HS48	3	5	10^{-4}	205
HS49	10	15	10^{-4}	420
HS50	6	10	10^{-5}	360
HS51	3	51	10^{-13}	213
HS52	>100		10^{4}	1293
HS53	3	6	10^{-13}	219
HS54	3	6	10^{-4}	219
HS55	5	11	10^{-7}	385
HS56	10	21	10^{-4}	418
HS6	>100		10^{1}	801
HS60	26	51	10^{-4}	536
HS61	6	27	10^{-4}	366
HS62	6	9	10^{-5}	310
HS63	7	15	10^{-4}	379
HS68	>100		10^{0}	963
HS69	>100		10^{3}	932
HS7	7	15	10^{-5}	335
HS77	>100		10^{-2}	1061
HS78	5	11	10^{-5}	370
HS79	15	31	10^{-4}	497
HS8	3	7	10^{-12}	170
HS80	7	15	10^{-4}	392
HS81	6	13	10^{-6}	388
HS9	>100		10^{0}	820
HS99	20	48	10^{-4}	457
HS99EXP	>100		10^{16}	3.4×10^{4}
HYDCAR6	>100		10^{8}	3.9×10^{4}
HYPCIR	3	20	10^{-6}	175
INTEGREQ	>100		10^{-2}	3×10^{4}
LCH	>100		10^{1}	10^{5}
LIN	*et*			
LOTSCHD	>100		10^{4}	2.7×10^{3}
LSNNODOC	9	17	10^{-7}	374
MARATOS	3	7	10^{-11}	165
METHANB8	*ov*			
METHANL8	>100		10^{113}	
MINPERM	3	7	10^{-11}	855
MSQRTA	>100		10^{-1}	1.3×10^{3}
MSQRTB	>100		10^{3}	2×10^{3}
ODFITS	5	10	10^{-4}	299
OPTCNTRL	>100		10^{-3}	
ORTHREGB	6	12	10^{-5}	444
ORTHREGC	>100		10^{1}	4.4×10^{3}
ORTHREGD	>100		10^{2}	4.5×10^{3}
ORTHREGE	>100		10^{2}	1.3×10^{3}
PFIT1	>100		10^{3}	1.2×10^{3}
PFIT2	>100		10^{4}	1.4×10^{3}
PFIT3	>100		10^{2}	10^{3}
PFIT4	>100		10^{4}	10^{3}
PORTFL1	2	4	10^{-13}	221
PORTFL2	1	2	10^{-5}	110
PORTFL3	2	4	10^{-11}	221
PORTFL4	1	2	10^{-5}	111
PORTFL6	2	4	10^{-15}	221
POWELLBS	>100		10^{4}	10^{3}
POWELLSQ	>100		10^{3}	929
READING5	>100		10^{6}	6.7×10^{4}
RECIPE	25	121	10^{-7}	462
RK23	>100		10^{2}	3×10^{3}
ROBOT	22	174	10^{-4}	666
SEMICON1	>100		10^{2}	2.4×10^{3}
SEMICON2	>100		10^{0}	2.5×10^{3}
TENBARS2	>100		10^{20}	
TENBARS3	>100		10^{20}	
TRIGGER	>100		10^{4}	1.4×10^{3}
ZANGWIL3	2	4	10^{-15}	128

Apêndice B

Teoremas e Lemas

Teorema B.1 *(Teorema 3.1.2 em [28]) Seja* $\|.\|$ *e* $|||.|||$ *duas normas em* $\mathbb{R}^n$. *Existem constantes positivas* α *e* β *tais que*

$$\alpha \|v\| \leq |||v||| \leq \beta \|v\|$$

para qualquer $v \in \mathbb{R}^n$.

Teorema B.2 *(Teorema 3.1.3 em [28]) Seja* $\|.\|$ *e* $|||.|||$ *duas normas em* $\mathbb{R}^{n\times n}$. *Existem constantes positivas* α *e* β *tais que*

$$\alpha \|A\| \leq |||A||| \leq \beta \|A\|$$

para qualquer $A \in \mathbb{R}^{n\times n}$.

Lema B.1 *(Lema 4.1.12 em [28]) Seja* $F : \mathbb{R}^n \to \mathbb{R}^m$ *continuamente diferenciável no conjunto aberto* $D \subset \mathbb{R}^n$, $x \in D$ *e seja* J *contínua à Lipschitz em* x *na vizinhança* D, *usando a norma de vector, a norma de operador de matriz induzida e a constante* γ. *Então, para qualquer* $x + p \in D$,

$$\|F(x+p) - F(x) - J(x)p\| \leq \frac{\gamma}{2} \|p\|^2 .$$

Lema B.2 *(Lema 4.1.15 em [28]) Considerem-se* F *e* J *satisfazendo as condições do Lema B.1. então, para todo* $v, u \in D$,

$$\|F(v) - F(u) - J(x)(v-u)\| \leq \gamma \frac{\|v-x\| + \|u-x\|}{2} \|v-u\|$$

Lema B.3 *(Lema 4.1.16 em [28]) Considerem-se F e J satisfazendo as condições do Lema B.1 e assuma-se que $J(x)^{-1}$ existe. Então existem $\epsilon > 0, 0 < \alpha < \beta$, tal que*

$$\alpha \|v - u\| \leq \|F(v) - F(u)\| \leq \beta \|v - u\|,$$

para todo $v, u \in D$, para os quais ve verifica $max\{\|v - x\|, \|u - x\|\} \leq \epsilon$.

Lema B.4 *(**Lema de Neumann** em [88]) Seja $B \in L(\mathbb{R}^n)$ e $\rho(B) < 1$. Então $(I-B)^{-1}$ existe e*

$$(I - B)^{-1} = \lim_{k \to \infty} \sum^{k} B^i.$$

Teorema B.3 *(**Teorema do valor médio** - Lema 3.3.10 em [88, pp 79]) Seja $f : D \subset \mathbb{R}^n \to \mathbb{R}$ e suponha-se que f admite segunda G-derivada em cada ponto do convexo $D_0 \subset D$. Então, para quaisquer x, $y \in D_0$, existe $t \in (0, 1)$ tal que*

$$f(x) - f(y) = -\nabla f(x)^T (y - x) - \frac{1}{2}(y - x)^T \nabla^2 f(x + t(y - x))(y - x)$$

Lema B.5 *(Lema 14.2.3 em [88, pp 480]) Seja $f : D \subset \mathbb{R}^n \to \mathbb{R}$ uma função contínua no aberto D e $G-$diferenciável no compacto $L^0 = L(f(x_0))$ para algum $x_0 \in D$. Para quaisquer $x \in L^0$ e $p \in \mathbb{R}^n$ com $\nabla f(x)^T p > 0$ existe um $\alpha^* > 0$ tal que*

$$f(x - \alpha^* p) = f(x) \quad e \quad [x, x - \alpha^* p] \subset L^0.$$

Em particular, se $\eta > 0$ for tal que

$$f(x - \alpha p) < f(x), \quad \forall x - \alpha p \in [x, x - \eta p] \cap L^0$$

então $[x, x - \eta p] \subset L^0$.

Lema B.6 *(Lema 14.1.5 em [88, pp 476]) Seja $f : D \subset \mathbb{R}^n \to \mathbb{R}$ e Ω o conjunto de pontos estacionários de f em D. Assuma-se que f é uma função continuamente diferenciável no compacto $D_0 \subset D$ e que Ω é finito. Considere-se uma qualquer sequência $\{x_k\} \subset D_0$ para a qual*

$$\lim_{k \to \infty} (x_k - x_{k+1}) = 0 \quad e \quad \lim_{k \to \infty} \nabla f(x_k) = 0.$$

Então

$$\lim_{k \to \infty} x_k = x^* \quad e \quad \nabla f(x^*) = 0$$

com $x^* \in \Omega$.

Teorema B.4 *(**Teorema de Taylor** em [87, pp 15]) Seja $f : \mathbb{R}^n \to \mathbb{R}$ continuamente diferenciável e $p \in \mathbb{R}^n$. Então tem-se que*

$$f(x+p) = f(x) + \nabla f(x+tp)^T p,$$

para algum $t \in (0,1)$. Se f for duas vezes continuamente diferenciável tem-se

$$\nabla f(x+tp) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p\, dt,$$

e

$$f(x+p) = f(x) + \nabla f(x)^T p + \tfrac{1}{2} p^T \nabla^2 f(x+tp)p$$

para algum $t \in (0,1)$.

Lema B.7 *(Lema 1.2.1 em [66]) Se M é uma matriz $n \times n$ com $\|M\| < 1$ então $I - M$ é uma matriz não singular e*

$$\left\|(I-M)^{-1}\right\| \leq \frac{1}{1-\|M\|}.$$

Teorema B.5 *(Teorema 1.2.1 em [66] - Teorema de Banach). Suponha-se $A,B \in \mathbb{R}^{n\times n}$ e B uma aproximação à inversa de A. Então A e B são ambas não singulares e*

$$\left\|A^{-1}\right\| \leq \frac{\|B\|}{1-\|I-BA\|}, \qquad \left\|B^{-1}\right\| \leq \frac{\|A\|}{1-\|I-BA\|},$$

e

$$\left\|A^{-1}-B\right\| \leq \frac{\|B\|\,\|I-BA\|}{1-\|I-BA\|}, \qquad \left\|A-B^{-1}\right\| \leq \frac{\|A\|\,\|I-BA\|}{1-\|I-BA\|}.$$

Teorema B.6 *(Teorema 10.1 em [85]) Seja $f(x) : \mathbb{R}^n \to \mathbb{R}$ definida num conjunto convexo aberto S. Suponha-se que $\nabla^2 f$ é contínua à Lipschitz em S, isto é,*

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leq L\|x-y\|, \ 0 < L < \infty;$$

para todo o x, y, em S e $0 < L < \infty$. Considere-se a sequência $\{x_k\}$ gerada por

$$x_{k+1} = x_k + p_k.$$

Suponha-se que $\{x_k\} \subset S$,

$$\lim_{k\to\infty} x_k = x^* \in S,$$

e $x_k \neq x^*$ *para todo o* k. *Assuma-se ainda que* $\nabla^2 f(x^*)$ *é definida positiva. Então* $\{x_k\}$ *converge superlinearmente para* x^* *e* $\nabla f(x^*) = 0$ *se e só se*

$$\lim_{k\to\infty} \frac{\|p_k - (p_N)_k\|}{\|p_k\|} = 0$$

em que $(p_N)_k$ *é a direcção Newton no ponto* x_k.

Teorema B.7 *(Teorema 10.2 em [85]) Seja* $f(x) : \mathbb{R}^n \to \mathbb{R}$. *Considerem-se* x_0 *uma aproximação inicial da sucessão* $\{x_k\}$ *definida por* $x_{k+1} = x_k + \alpha_k p_k$, *em que* $p_k \in \mathbb{R}^n$ *e* α_k *é um escalar. Suponha-se que i)* $S = \{x : f(x) \leq f(x_0)\}$ *é limitado; ii)* ∇f *é contínua à Lipschitz para todo o* $x, y \in S$, *isto é,*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \ 0 < L < \infty;$$

iii) O vector p_k *satisfaz a condição de suficientemente descendente ,*

$$-\frac{p_k^T \nabla f(x_k)}{\|p_k\| \, \|\nabla f(x_k)\|} \geq \epsilon > 0;$$

iv) As direcções de procura p_k *são do tipo gradiente:*

$$\|p_k\| \geq m \|\nabla f(x_k)\|, \text{ para todo o } k \text{ (com } m \text{ constante positiva)}$$

e são limitadas em norma:

$$\|p_k\| \leq M, \text{ para } M \text{ constante e para todo o } k;$$

v) O escalar α_k *é escolhido como o primeiro elemento da sequência* $\left\{1, \frac{1}{2}, \frac{1}{4}, ...\right\}$ *a satisfazer a condição de redução significativa*

$$f(x_k + \alpha p_k) \leq \ f(x_k) + \mu \alpha_k p_k^T \nabla f(x_k), \ 0 < \mu < 1.$$

Então

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Lema B.8 *(Lema 3.1 em [36]) Seja* $P(x) : \mathbb{R}^n \to \mathbb{R}$ *uma função duas vezes continuamente diferenciável numa vizinhança de* $\overline{x}$ *tal que* $\nabla P(\overline{x}) = 0$. *Suponha que, para todo o* k

$$\nabla P(x_k)^T d_k \leq -\gamma \|d_k\|^2$$

para algum $\gamma > 0$. *Então, para qualquer* $\mu \in (0, \frac{1}{2})$, *existe um* $\overline{k}$ *tal que, para todo o* $k \geq \overline{k}$

$$P(x_k + d_k) \leq P(x_k) + \mu \nabla P(x_k)^T d_k.$$

Teorema B.8 *(Teorema em [108, pp 306]) Se*

1. $\{\mathbb{R}^n, \|.\|\}$ é um espaço linear normado;
2. $S \subset \mathbb{R}^n$ é compacto;
3. $\{x_k\}$ é uma sequência em S;
4. x^* é o único ponto limite de $\{x_k\}$;

então $\{x_k\}$ converge para x^* quando $k \to \infty$.

Bibliografia

[1] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: Building blocks for iterative methods*, SIAM, 1994.

[2] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear programming theory and algorithms*, John Wiley & Sons, Inc., 1993.

[3] H.Y. Benson, D.F. Shanno, and R.J. Vanderbei, *Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions*, Tech. report, Operations Research and Financial Engineering, Princeton University, 2000.

[4] I. Bongartz, A.R. Conn, N.I.M. Gould, and Ph.L. Toint, *CUTE: Constrained and Unconstrained Testing Environment*, ACM Transaction on Mathematical Software (1995), no. 21, 123–160.

[5] D. Boukari and A.V. Fiacco, *Survey of penalty, exact-penalty and multiplier methods from 1968 to 1993*, Optimization **22** (1995), 301–334.

[6] C.G. Broyden, J.E. Dennis, and J.J. Moré, *On the local and superlinear convergence of quasi-Newton methods*, Journal of the Institute of Mathematics and its Applications **12** (1973), 223–245.

[7] J.P. Bulteau and J.P. Vial, *A restricted trust region algorithm for unconstrained optimization*, Journal of Optimization Theory and Applications **47** (1985), no. 4, 413–435.

[8] J.V. Burke, *An exact penalization viewpoint of constrained optimization*, SIAM Journal on Control and Optimization **29** (1991), 968–998.

[9] R. Byrd and R. Schnabel, *Continuity of the null space basis and constrained optimization*, Mathematical Programming **35** (1986), 32–41.

[10] Richard H. Byrd and Jorge Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM Journal on Numerical Analysis **26** (1989), no. 3, 727–739.

[11] M.I.R.F. Caiado, *Caracterização e estudo de algumas propriedades do método de Newton modificado*, Master's thesis, Universidade do Minho, Janeiro 1999.

[12] M.I.R.F. Caiado and E.M.G.P. Fernandes, *Modified Newtont's method : Properties and convergence issues*, Tech. Report ALG / EF / 2, Algoritmi R & D Center, Systems Engineering Group, Universidade do Minho, Outubro 1999.

[13] M.R. Celis, J.E. Dennis, and R.A. Tapia, *A trust-region strategy for nonlinear equality constrained optimization*, Numerical Optimization 84 (R. Byrd P. Boggs and R. Schnabel, eds.), SIAM, 1985, pp. 71–82.

[14] R.M. Chamberlain, M.J.D. Powell, C. Lemarechal, and H.C. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Mathematical Programming **16** (1982), 1–17.

[15] D. Conforti, L. Grandinetti, and R. Musmanno, *A paralell tensor algorithm for nonlinear optimization*, Optimization Methods and Software **3** (1994), 125–142.

[16] A.R. Conn, N. Gould, and Ph.L. Toint, *Large-scale nonlinear constrained optimization: A current survey*, Algorithms for Continuous Optimization: State of the Art (1994).

[17] A.R. Conn, N.I.M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM Journal on Numerical Analysis **25** (1988), no. 2, 433–460.

[18] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, *LANCELOT: A fortran package for large-scale nonlinear optimization (Release A)*, Springer Series in Computational Mathematics, 17, Springer-Verlag, Heidelberg, Berlin, 1992.

[19] R.S. Dembo, S.C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis **19** (1982), no. 2, 400–408.

[20] R.S. Dembo and T. Steihaug, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Mathematical Programming **26** (1983), 190–212.

[21] J.W. Demmel, *Applied numerical linear algebra*, SIAM, 1997.

[22] N. Deng and Z. Li, *Global convergence of three terms conjugate gradient methods*, Optimization Methods and Software **4** (1995), 273–282.

[23] J.E. Dennis, M. El-Alem, and M.C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM Journal on Optimization **7** (1997), no. 1, 177–207.

[24] J.E. Dennis, M. El-Alem, and K. Williamson, *A trust-region approach to nonlinear systems of equalities and inequalities*, SIAM Journal on Optimization **9** (1999), no. 2, 291–315.

[25] J.E. Dennis, M. Heinkenschloss, and L.N. Vicente, *Trust-region interior-point SQP algorithms for a class of nonlinear programming problems*, SIAM Journal on Control and Optimization **36** (1998), no. 5, 1750–1794.

[26] J.E. Dennis and J.J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Mathematics of Computation **28** (1974), no. 126, 549–560.

[27] J.E. Dennis and R.B. Schnabel, *A view of unconstrained optimization*, Optimization (A.H.G. Rinnooy Kan G.L. Nemhauser and M.J. Todd, eds.), North-Holland, 1989, pp. 1–72.

[28] ———, *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, 1996.

[29] J.E. Dennis and L.N. Vicente, *On the convergence theory of trust-region-based algorithms for equality-constrained optimization*, SIAM Journal on Optimization **7** (1997), no. 4, 927–950.

[30] L.C.W. Dixon, *On Nazaretht's three term conjugate gradient method*, Tech. Report 133, The Hatfield Polytechnic, 1983.

[31] L.C.W. Dixon, P.G. Ducksbury, and P. Singh, *A new three-term conjugate gradient method*, Journal of Optimization Theory and Applications **47** (1985), no. 3, 285–300.

[32] S.C. Eisenstat and H.F. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM Journal on Scientific Computation **17** (1996), 16–32.

[33] M. El-Alem, *A global convergence theory for the Celis-Dennis-Tapia trust-region algorithm for constrained optimization*, SIAM Journal on Numerical Analysis **28** (1991), no. 1, 266–290.

[34] J. Eriksson, *A curvilinear search for systems of nonlinear equations*, http://www.cs.umu.se/ jerry/research.html.

[35] F. Facchinei, J. Júdice, and J. Soares, *An active set Newton algorithms for large-scale nonlinear programs with box constraints*, SIAM Journal on Optimization **8** (1998), no. 1, 158–186.

[36] F. Facchinei and S. Lucidi, *Local properties of a Newton-like direction for equality constrained minimization problems*, Optimization Methods and Software **3** (1994), 13–26.

[37] E.M.G.P. Fernandes, *An exact penalty approach with Newton based descent directions for nonlinear optimization with constraints*, 5th Stockholm Optimization Days, 1994.

[38] ———, *A Curry-Altman-Armijo line search algorithm which uses Newton based descent directions*, Proceedings of APMODt'95 (London), 1995, pp. 27–30.

[39] ———, *Newton based exact penalty techniques for nonlinear optimization with constraints*, Operations Research Proceedings 1994 (Berlin) (A. Bachem and A. Drexl, eds.), Springer-Verlag, 1995, pp. 39–44.

[40] M.C. Ferris, S. Lucidi, and M. Roma, *Nonmonotone curvilinear line search methods for unconstrained optimization*, Computational Optimization and Applications **6** (1996), 117–136.

[41] A.V. Fiacco and G.P. McCormick, *Nonlinear programming - sequential unconstrained minimization techniques*, SIAM, 1990.

[42] R. Fletcher, *A class of methods for nonlinear programming with termination and convergence properties*, Integer and Nonlinear Programming (1970), 157–175.

[43] ———, *Practical methods of optimization, 2nd edition*, John Wiley & Sons, Ltd, 1987.

[44] R. Fletcher, N.I.M. Gould, S. Leyffer, and Ph.L. Toint, *Global convergence of trust-region SQP-filter algorithms for general nonlinear programming*, Tech. Report 99/03, Département de Mathématique, FUNDP, Namur, 1999.

[45] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Tech. Report NA/171, University of Dundee, Department of Mathematics, 1998.

[46] R. Fletcher, S. Leyffer, and Ph.L. Toint, *On the global convergence of an SLP-filter algorithm*, Tech. Report 98/13, Département de Mathématique, FUNDP, Namur, 1998.

[47] R. Fontecilla, *On inexact quasi-Newton methods for constrained optimization*, SIAM Numerical Optimization (1984), 102–118.

[48] ———, *Inexact secant methods for nonlinear constrained optimization*, SIAM Journal on Numerical Analysis **27** (1990), no. 1, 154–165.

[49] A. Forsgreen and P.E. Gill, *Primal-dual interior methods for nonlinear programming*, SIAM Journal on Optimization **8** (1998), no. 4, 1132–1152.

[50] L. Fox and D.F. Meyers, *Computing methods for scientists and engineers*, Clarendon Press, 1968.

[51] P. Gill, W. Murray, M. Saunders, and M. Wright, *NPSOL: A fortran package for nonlinear programming*, Stanford Office of Technology Licensing, Califórnia, USA, 1986.

[52] Ph.E. Gill and W. Murray, *Safeguarded steplength algorithms for optimization using descent directions*, Tech. report, National Physical Laboratory, Division of Numerical Analysis and Computing, Teddington, 1974.

[53] Ph.E. Gill, W. Murray, and M.H. Wright, *Practical optimization*, Academic Press, 1981.

[54] T. Glad and E. Polak, *A multiplier method with automatic limitation of penalty growth*, Mathematical Programming **17** (1979), 140–155.

[55] D. Goldfarb, *Curvilinear path steplenght algorithms for minimization which use directions of negative curvature*, Mathematical Programming **18** (1980), 31–40.

[56] G.H. Golub and D.P. O'Leary, *Some history of the conjugate gradient and Lanczos algorithms: 1948-1976*, SIAM Review **31** (1989), no. 1, 50–102.

[57] N.I.M. Gould, S. Lucidi, M. Roma, and Ph.L. Toint, *A linesearch algorithm with memory for unconstrained optimization*, High Performance Algorithm and Software in Nonlinear Optimization (Renato De Leone, Almerico Murli, Panos M. Pardalos, and Gerardo Toraldo, eds.), Kluwer Academic Publishers, 1998, pp. 207–223.

[58] ———, *Solving the trust-region subproblem using the Lanczos method*, SIAM Journal on Optimization **9** (1999), no. 2, 504–525.

[59] ———, *Exploiting negative curvature directions in linesearch metods for unconstrained optimization*, November 25, 1997.

[60] A. Greenbaum, *Iterative methods for solving linear systems*, SIAM, 1997.

[61] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone linesearch technique for Newtont's method*, SIAM Journal on Numerical Analysis **23** (1986), 707–716.

[62] ———, *A class of nonmonotone stabilization methods in unconstrained optimization*, Numerical Mathematical **59** (1991), 779–805.

[63] S.P. Han, *Exact penalty functions in nonlinear programming*, Mathematical Programming **17** (1979), 251–269.

[64] M. Heinkenschloss and L.N. Vicente, *Analysis of inexact trust-region interior-point SQP algorithms*, Tech. Report TR95-18, Rice University - Department of Computational & Applied Mathematics, Junho 1995.

[65] J. Jahn, *Introduction to the theory of nonlinear optimization*, Springer-Verlag, 1994.

[66] C.T. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, 1995.

[67] ———, *Iterative methods for optimization*, SIAM, 1999.

[68] A. Liao, *A reduced hessian method for constrained optimization*, Computational Optimization and Applications **2** (1993), 129–143.

[69] S. Lucidi, *New results on a continuously differentiable exact penalty function*, SIAM Journal on Optimization **2** (1992), 558–574.

[70] S. Lucidi, F. Rochetich, and M. Roma, *Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization*, SIAM Journal on Optimization **8** (1998), no. 4, 916–939.

[71] S. Lucidi and M. Roma, *Numerical experiences with new truncated Newton methods in large scale unconstrained optimization*, Computational Optimization and Applications **7** (1997), 71–87.

[72] H. Lutkepohl, *Handbook of matrices*, John Wiley & Sons, 1996.

[73] O.L. Mangasarian, *Nonlinear programming*, SIAM, 1994.

[74] The MathWorks, Inc., *MATLAB - optimization toolbox*, 1999.

[75] G.P. McCormick, *A modification of Armijo's step-size rule for negative curvature*, Mathematical Programming **13** (1977), 111–115.

[76] M.T. Monteiro and E.M. Fernandes, *Método de substituição do vector dos multiplicadores baseado em pseudo-derivadas*, Investigação Operacional **17** (1997), no. 1, 87–100.

[77] M.T.T. Monteiro, *Técnica de penalização baseada na substituição do vector dos multiplicadores com procura curvilínea*, Master's thesis, Universidade do Minho, Junho 1996.

[78] M.T.T. Monteiro and E.M.G.P. Fernandes, *A combined modification of Newtont's method for systems of nonlinear equations*, Proceedings of "Copper Mountain Conference on Iterative Methods", Cray Research Inc. Colorado, 1996, pp. 1–7.

[79] ———, *A curvilinear search approach - comparison with MATLAB and NPSOL*, Actas Do IV Congreso Galego de Estatística e Investigación de Operacións (Santiago de Compostela), 1999, pp. 103–106.

[80] ———, *A curvilinear pseudo-Newton algorithm for nonlinear programming*, Operations Research Proceedings 1999 (K. Inderfurth, G. Schwodiauer, W. Domschke, F. Juhnke, P. Kleinschmidt, and G. Wascher, eds.), Springer-Verlag, Berlin, 2000, pp. 23–29.

[81] J.J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, Numerical Analysis (G.A. Watson, ed.), Springer-Verlag, 1977, pp. 105–116.

[82] J.J Moré, B.S. Garbow, and K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software **7** (1981), no. 1, 17–41.

[83] J.J. Moré and D.C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Mathematical Programming **16** (1979), 1–20.

[84] J.J. Moré and S.J. Wright, *Optimization software guide*, SIAM, 1993.

[85] S.G. Nash and A. Sofer, *Linear and nonlinear programming*, McGraw-Hill International Editions, 1996.

[86] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numérica (1992), 199–242.

[87] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer-Verlag, 1999.

[88] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, 1970.

[89] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis **12** (1975), no. 4, 617–629.

[90] E.R. Panier and A. Tits, *Avoiding the Maratos effect by means of a nonmonotone line search i. general constrained problems*, SIAM Journal on Numerical Analysis **28** (1991), no. 4, 1183–1195.

[91] E.R. Panier and A.L. Tits, *On combining feasibility, descent and superlinear convergence in inequality constrained optimization*, Mathematical Programming **59** (1993), 261–276.

[92] G. Di Pillo and L. Grippo, *An exact penalty method with global convergence properties for nonlinear programming problems*, Mathematical Programming **36** (1986), 1–18.

[93] ———, *Exact penalty functions in constrained optimization*, SIAM Journal on Control and Optimization **27** (1989), 1333–1360.

[94] W.C. Rheinboldt, *Methods for solving systems of nonlinear equations*, SIAM, Philadelphia, 1974.

[95] Y. Saad and M. Schultz, *Conjugate-gradient like algorithms for solving nonsymmetric linear systems*, Mathematics of Computation **44** (1985), 417–424.

[96] A. Sartenaer, *Automatic determination of an initial trust region in nonlinear programming*, SIAM Journal on Scientific Computing **18** (1997), no. 6, 1788–1803.

[97] K. Schittkowsky, *More test examples for nonlinear programming codes*, Springer-Verlag, 1987.

[98] D.C. Sorensen, *Newtont's method with a model trust region modification*, SIAM Journal on Numerical Analysis **19** (1982), no. 2, 409–426.

[99] E. Spedicato, *A review of ABS algorithms for general linear systems*, Optimization Methods and Software **1** (1992), 171–182.

[100] Y. Tanaka, *A trust-region method for semi-infinite programming problem*, International Journal of Systems Science **30** (1999), no. 2, 199–204.

[101] R.A. Tapia, *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, Journal of Optimization Theory and Applications **22** (1977), 135–194.

[102] L.N. Trefethen and D. Bau III, *Numerical linear algebra*, SIAM, 1997.

[103] M. Ulbrich, S. Ulbrich, and L.N. Vicente, *A globally convergent primal-dual interior-point filter method for nonconvex nonlinear programming*, Tech. Report 00-11, Universidade de Coimbra, Departamento de Matemática, 2000.

[104] R.J. Vanderbei, *LOQO: An interior point code for quadratic programming*, Tech. Report SOR-94-15, Statistics and Operations Research, Princeton University, 1998.

[105] R.J. Vanderbei and D.F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications **13** (1999), 231–252.

[106] L.N. Vicente, *What happens when we trust a region that is a line?*, Tech. Report TR95-10, Rice University - Department of Computational & Applied Mathematics, Abril 1995.

[107] ———, *A comparison between line searches and trust regions for nonlinear optimization*, Investigação Operacional **16** (1996), no. 2, 173–179.

[108] M.A. Wolfe, *Numerical methods for unconstrained optimization*, Van Nostrand Reinhold Company, 1978.

[109] Y. Yuan, *On the truncated conjugate gradient method*, Mathematical Programming **A 87** (2000), 561–573.

[110] D. Zhu, *A nonmonotone line search technique for improved projected quasi-Newton methods*, Optimization **18** (1993), 31–45.