

PEDRO JORGE SEQUEIRA CARDOSO

MÉTODOS DE SUBSTITUIÇÃO
DO VECTOR DOS MULTIPLICADORES
BASEADOS EM ACTUALIZAÇÕES
QUASI-NEWTON

UNIVERSIDADE DO MINHO
DEPARTAMENTO DE MATEMÁTICA
FEVEREIRO DE 1999

PEDRO JORGE SEQUEIRA CARDOSO

MÉTODOS DE SUBSTITUIÇÃO DO VECTOR DOS MULTIPLICADORES

BASEADOS EM ACTUALIZAÇÕES QUASI-NEWTON

*Dissertação de mestrado em Matemática
Computacional apresentado à Universidade do
Minho, para obtenção do grau de mestre.*

-
- À minha orientadora, Doutora Edite Manuela da G.P. Fernandes, por me ter proposto o tema deste trabalho, pela sua orientação, pela seu incentivo, permanente disponibilidade, leitura cuidadosa do manuscrito e consequente crítica construtiva.

 - À Escola E.B.2,3/S Dr. Daniel de Matos-Poiães, pelas facilidades concedidas durante a parte lectiva.

 - À Universidade Católica Portuguesa.

 - À Área Departamental de Engenharia Eléctrica Electrónica da Escola Superior de Tecnologia da Universidade do Algarve, pelos apoios concedidos.

 - A todos os que, mesmo indirectamente, me apoiaram durante este período.
-

À minha esposa Cláudia
e aos meus pais Maria e João.

Resumo

Neste trabalho recorre-se a técnicas de penalização exacta baseadas na função Lagrangeana aumentada, para a resolução de um problema de optimização não linear com restrições de igualdade. O processo requer o uso de estimativas do vector dos multiplicadores de Lagrange, presente na Lagrangeana aumentada, e é baseado numa filosofia de substituição desse vector. Num dos casos, a estimativa é uma função do vector das restrições e, no outro, depende do gradiente da função objectivo e da inversa generalizada do Jacobiano das restrições.

A resolução do problema sem restrições recorre a técnicas conhecidas. Neste contexto, foram implementadas cinco versões: uma versão Newton, uma Pseudo-Newton, que usa aproximações às derivadas, uma Quasi-Newton e duas versões que combinam o método de Newton com aproximações Quasi-Newton.

A escolha do parâmetro de penalização, na minimização sem restrições, foi feita tendo como base uma função linear que aumenta com a violação das restrições, mas é inferior e superiormente limitada.

Finalmente, são apresentados os resultados computacionais obtidos e algumas conclusões.

Abstract

In this work an exact penalty technique based on an augmented Lagrangian function is used to solve a nonlinear equality constrained minimization problem. The process requires Lagrangian multiplier vector estimates and it is based on a vector substitution philosophy. In one case, the estimate is a function of the constraints alone and in the other it requires the use of the gradient of the objective function and of the generalized inverse of the Jacobian of the constraints.

The solution of the unconstrained problem resolution uses known techniques. In this context, five versions were implemented: a Newton version, a Pseudo-Newton, that uses derivative approximations, a Quasi-Newton and two versions which combine Newton's method with Quasi-Newton approximations.

The penalty parameter for the unconstrained minimization was taken as a linear function which increases with the constraint violation but has inferior and superior bounds.

Finally, computational results and conclusions are presented.

Índice Geral

Resumo	i
Abstract	iii
1 Introdução	1
2 Definições e Resultados Básicos	5
2.1 Álgebra Linear	5
2.1.1 Matrizes	7
2.1.2 Valores e Vectores Próprios	9
2.2 Análise	9
2.3 Síntese	12
3 Sistemas de Equações não Lineares	13
3.1 Método de Newton	14
3.2 Método Quasi-Newton. Equação de Broyden	16
3.3 Síntese	22
4 Optimização Sem Restrições	23
4.1 Condições de Optimalidade	23
4.2 Método de Newton	26
4.3 Métodos Quasi-Newton	31
4.3.1 Fórmulas de Actualização Para a Hessiana	34
4.3.2 Fórmulas de Actualização Para a Inversa da Hessiana	38
4.4 Síntese	42
5 Optimização Com Restrições	43
5.1 Multiplicadores de Lagrange	43
5.2 Funções de Penalização	47

5.2.1	Função de Penalização Quadrática	47
5.2.2	Função Lagrangeana Aumentada	49
5.3	Técnicas de Penalização Exacta	50
5.3.1	Fórmula de Substituição $\lambda(x) = -\rho c(x)$	51
5.3.2	Fórmula de Substituição $\lambda(x) = (\nabla c(x)^+)^T \nabla f(x)$	53
5.3.3	Recurso a Pseudo-Derivadas	57
5.4	Síntese	59
6	Experiências Computacionais	61
6.1	Implementação	61
6.1.1	Versão Newton Simples	61
6.1.2	Versão Quasi-Newton(BFGS)	62
6.1.3	Versão Pseudo-Newton	63
6.1.4	Versão Newton/Quasi-Newton(BFGS)	63
6.1.5	Versão Newton/Quasi-Newton(BFGS/Broyden)	64
6.2	Critério de Paragem	66
6.3	Parâmetro de Penalização	66
6.4	Resultados Computacionais	67
6.5	Síntese	98
7	Conclusões e Trabalho Futuro	101
7.1	Conclusões	101
7.2	Trabalho Futuro	104
	Bibliografia	104

Capítulo 1

Introdução

O problema de optimização é, de modo geral, um problema onde se pretende calcular o mínimo (ou máximo) de uma dada função de várias variáveis, estando estas sujeitas a um conjunto de restrições. O problema pode então ser colocado matematicamente, do seguinte modo

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.a.} \quad & \\ & c_i(x) = 0, \quad i = 1, \dots, m' \\ & c_i(x) \geq 0, \quad i = m' + 1, \dots, m \end{aligned} \tag{1.1}$$

em que $f, c_i : \mathbb{R}^n \rightarrow \mathbb{R}$, ($i = 1, \dots, m$). A f chamamos *função objectivo* e ao conjunto $\{c_i : i \in \{1, \dots, m\}\}$ chamamos o conjunto das *funções de restrição*. Ao conjunto dos pontos, x , que verificam as restrições chamamos *conjunto admissível*. Assim, resolver o problema (1.1) consiste em calcular os pontos admissíveis para os quais a função objectivo assume um valor mínimo.

O facto da maioria dos problemas se poder colocar na forma (1.1), não significa que se deva ignorar diferenças entre problemas. Perante um problema, é geralmente vantajoso determinar algumas das suas características especiais, que permitam resolver o problema de forma mais eficiente. Por exemplo, poderá ser possível omitir testes a situações que não podem ocorrer, ou mesmo, evitar recalcular quantidades que não variam.

As tabelas seguintes dão um esquema de classificação baseado nas características das funções do problema, donde se podem obter vantagens algorítmicas significativas. Assim, relativamente à função objectivo temos:

Propriedades da função objectivo
Função de uma única variável
Função linear
Função quadrática
Soma de quadrados de funções lineares
Soma de quadrados de funções não lineares
Função não linear suave
Função não linear esparsa
Função não linear não suave

Quanto às restrições temos:

Propriedades das funções de restrição
Sem restrições
Limites simples
Funções lineares
Funções lineares esparsas
Funções não lineares suaves
Funções não lineares esparsas
Funções não lineares não suaves

Há outras características importantes a distinguir nos problemas de optimização. O tamanho do problema afecta tanto a nível de armazenamento de dados como do esforço computacional requeridos para obter a solução, pelo que técnicas eficazes para problemas de pequena dimensão não são, em geral, aceitáveis para problemas de grandes dimensões. Outra característica que varia de problema para problema, envolve a informação que está disponível para o algoritmo durante o processo. Por exemplo, num caso pode ser possível calcular analiticamente as primeiras e segundas derivadas das funções, enquanto que, noutra caso, só os valores das funções objectivo e das restrições podem ser obtidos.

Este trabalho tem como objectivo implementar técnicas de penalização exacta, baseadas nos métodos de Newton, Quasi-Newton e numa combinação de ambos, para resolver problemas de optimização com restrições de igual-

dade específicos, nomeadamente com função objectivo e funções de restrição não lineares e com derivadas parciais contínuas até, pelo menos, à segunda derivada. Assim, este trabalho está dividido em sete capítulos, começando com este primeiro capítulo introdutório.

No segundo capítulo são indicados alguns resultados básicos de Álgebra Linear, principalmente ao nível de resultados matriciais, e definições de Análise.

Nos capítulos três e quatro são introduzidos os métodos de Newton e Quasi-Newton para resolução de sistemas de equações não lineares e de problemas de optimização sem restrições, respectivamente. Como veremos, estes capítulos estão relacionados entre si.

No capítulo cinco, é introduzido o problema de minimização com restrições de igualdade e indicada a técnica usada, neste trabalho, para o resolver - a técnica de penalização exacta.

No sexto capítulo são apresentadas as cinco versões que foram implementadas para comparação computacional.

Finalmente, no capítulo sete são referidas as conclusões retiradas deste trabalho e apontadas áreas de investigação futura com o objectivo de melhorar o desempenho dos métodos de substituição.

Capítulo 2

Definições e Resultados Básicos

Neste capítulo são introduzidas algumas definições e resultados básicos de Álgebra Linear e Análise necessários para os restantes capítulos.

2.1 Álgebra Linear

Esta secção descreve os conceitos de Álgebra Linear mais importantes para a optimização.

Um vector de \mathbb{R}^n pode representar-se da seguinte forma

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1, x_2, \dots, x_n]^T.$$

Dados $x, y \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}$, podemos definir as operações usuais de adição de vectores

$$x + y = [x_1 + y_1, \dots, x_n + y_n]^T \quad (2.1)$$

e multiplicação de vectores por um escalar

$$\alpha x = [\alpha x_1, \dots, \alpha x_n]^T. \quad (2.2)$$

O conjunto \mathbb{R}^n com as operações definidas em (2.1) e (2.2) constitui um *espaço vectorial*.

Definição 2.1 A qualquer operação, $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$, que satisfaça as seguintes propriedades:

1. $\forall x \in \mathbb{R}^n : \|x\| \geq 0$ e $\|x\| = 0$ se e só se $x = 0$
2. $\forall \alpha \in \mathbb{R} \forall x \in \mathbb{R}^n : \|\alpha x\| = |\alpha| \|x\|$
3. $\forall x, y \in \mathbb{R}^n : \|x + y\| \leq \|x\| + \|y\|$ (Desigualdade Triangular)

chamamos *norma vectorial*.

As normas mais conhecidas vêm em função da *norma-p* que é definida por

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}. \quad (2.3)$$

Os três valores mais comuns para p são $p = 1, 2$ e ∞ , cujas normas correspondentes são denominadas por *norma-um*, *norma-dois* ou *norma Euclidiana* e *norma-infinito*. Note-se que $\|x\|_2^2 = x^T x$ e que quando $p = \infty$ vem $\|x\|_\infty = \max_i |x_i|$. Um espaço vectorial considerado com uma norma é chamado *espaço vectorial normado*.

Definição 2.2 Dados dois vectores $x, y \in \mathbb{R}^n$, define-se *produto interno (ou escalar) usual* por

$$x|y = x^T y = \sum_{i=1}^n x_i y_i. \quad (2.4)$$

Os vectores x e y dizem-se *ortogonais* se $x^T y = 0$.

Quaisquer que sejam $x, y \in \mathbb{R}^n$ verifica-se $|x^T y| \leq \|x\|_2 \|y\|_2$, conhecida como *desigualdade de Schwarz*.

Definição 2.3 Se $x^T x = 1$ então x diz-se *normalizado*. O conjunto de vectores $\{x_i \in \mathbb{R}^n : i = 1, 2, \dots, m \leq n\}$ diz-se *ortonormal* se

$$x_i^T x_j = \delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}.$$

Uma norma matricial pode ser definida em termos de uma norma vectorial do seguinte modo: dada uma norma vectorial $\|\cdot\|$ e uma matriz A , considere-se

$$\|A\| = \max_{\|x\|=1} \|Ax\|.$$

As três normas vectoriais já definidas induzem três normas matriciais para uma matriz $A_{m \times n} = [a_{ij}]_{m \times n}$:

$$\begin{aligned}\|A\|_1 &= \max_{1 \leq j \leq n} \left(\sum_{i=1}^m |a_{ij}| \right) \\ \|A\|_2 &= \left(\lambda_{\max}[A^T A] \right)^{\frac{1}{2}} \\ \|A\|_\infty &= \max_{1 \leq i \leq m} \left(\sum_{j=1}^n |a_{ij}| \right)\end{aligned}$$

A norma-um é o máximo da soma dos valores absolutos por coluna, a norma-dois é a raiz quadrada do maior valor próprio da matriz $A^T A$ e a norma-infinito é o máximo da soma dos valores absolutos por linha. Outra norma importante é a *norma de Frobenius*, denotada por $\|\cdot\|_F$, que corresponde a considerar uma matriz $A_{m \times n}$ como um vector com nm elementos, e depois calcular a sua norma Euclidiana:

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{\frac{1}{2}}$$

2.1.1 Matrizes

Nesta secção, são apresentadas algumas definições e resultados que serão utilizados nos restantes capítulos, podendo as demonstrações ser encontradas em vários livros de Álgebra Linear.

Definição 2.4 Dada uma matriz quadrada A dizemos que ela é *simétrica* se e só se $A^T = A$. Dizemos que é *anti-simétrica* se e só se $A^T = -A$. Caso $AA^T = I$ então A diz-se *ortogonal*

Teorema 2.1 Dadas duas matrizes quadradas A e B , verificam-se as seguintes propriedades:

1. $(AB)^T = B^T A^T$;
2. Se A e B são não singulares então $(AB)^{-1} = B^{-1} A^{-1}$;
3. $\det(AB) = \det(A) \det(B)$;
4. $\det(A^T) = \det(A)$;
5. Se A é uma matriz singular então $\det(A) = 0$;

6. Dado o vector x então

$$x^T Ax = \sum_{i,j=1}^n x_i a_{ij} x_j = \sum_{i,j=1}^n x_i a_{ji} x_j;$$

7. Se $B = \frac{1}{2}(A+A^T)$ (B é uma matriz simétrica) então $x^T Ax = x^T Bx$.

Definição 2.5 Uma matriz quadrada A é *positiva definida* (*positiva semi definida*) se todos os seus valores próprios são estritamente positivos (não negativos). De forma análoga podemos definir *negativa definida* (*negativa semi definida*). Se uma matriz não verificar nenhuma das condições anteriores diz-se *indefinida*.

De uma forma equivalente, temos:

Definição 2.6 Dada uma matriz A , ela é

1. *positiva definida* se $x^T Ax > 0, \forall x \in \mathbb{R}^n, x \neq 0$;
2. *positiva semi definida* se $x^T Ax \geq 0, \forall x \in \mathbb{R}^n$;
3. *negativa definida* se $x^T Ax < 0, \forall x \in \mathbb{R}^n, x \neq 0$;
4. *negativa semi definida* se $x^T Ax \leq 0, \forall x \in \mathbb{R}^n$;

Temos ainda alguns resultados importantes:

Teorema 2.2 1. Se A é uma matriz positiva definida então $a_{ii} > 0, i = 1, 2, \dots, n$. Resultados semelhantes verificam-se quando A é positiva semi definida, negativa definida e negativa semi definida.

2. Se A é simétrica e definida positiva, então existe uma matriz M , também simétrica e definida positiva, tal que $A = M^2$.

Podemos assim definir os seguintes conjuntos de matrizes:

$$\begin{aligned} PD &= \{A : x^T Ax > 0\} \\ PSD &= \{A : x^T Ax \geq 0\} \\ ND &= \{A : x^T Ax < 0\} \\ NSD &= \{A : x^T Ax \leq 0\} \end{aligned}$$

2.1.2 Valores e Vectores Próprios

Definição 2.7 Para cada matriz quadrada, existe pelo menos um λ e um correspondente vector u tal que

$$Au = \lambda u, \quad (2.5)$$

isto é, o vector transformado é apenas um múltiplo do vector original. Ao conjunto dos escalares λ que satisfazem (2.5) chamamos *valores próprios* e aos vectores u chamamos *vectores próprios*.

Alguns resultados importantes são indicados no seguinte teorema:

Teorema 2.3

1. Se A for uma matriz simétrica de ordem n então todos os valores próprios de A são reais e tem n valores próprios distintos;
2. Se A for não singular então todos os seus valores próprios são não nulos e os valores próprios de A^{-1} são os recíprocos dos de A ;
3. Se a matriz A for simétrica e não singular então os valores próprios de A^{-1} são reais e distintos;
4. Se $\Pi(A)$ representa o produto dos valores próprios de A e A e B são matrizes quadradas quaisquer, então

$$\Pi(AB) = \Pi(A)\Pi(B).$$

2.2 Análise

Devido à essência do problema que se pretende resolver, definido em (1.1), existe a necessidade de introduzir outras noções e resultados importantes.

Consideremos o espaço linear normado $\{\mathbb{R}^n, \|\cdot\|\}$, com uma das normas de vectores definidas em 2.1. À custa dessas normas podemos ainda definir *distância* entre dois pontos $x, y \in \mathbb{R}^n$, considerando a função

$$d(x, y) = \|x - y\|$$

Para generalizar as ideias de intervalos fechados e abertos em \mathbb{R} , comecemos por definir bola aberta e fechada em $\{\mathbb{R}^n, \|\cdot\|\}$.

Definição 2.8 Dado $r \in \mathbb{R}_0^+$ e um ponto $x \in \mathbb{R}^n$ chamamos:

- *bola fechada de centro x e raio r* ao conjunto

$$B[x, r] = \{y \in \mathbb{R}^n : \|x - y\| \leq r\};$$

- *bola aberta de centro x e raio r* ao conjunto

$$B(x, r) = \{y \in \mathbb{R}^n : \|x - y\| < r\}.$$

Definição 2.9 A sequência $\{x^{(k)}\}$ num espaço linear normado $\{\mathbb{R}^n, \|\cdot\|\}$ converge para $x^* \in \mathbb{R}^n$ se e só se

$$\lim_{k \rightarrow \infty} \|x^{(k)} - x^*\| = 0.$$

Dizemos que $\{x^{(k)}\}$ é uma *sequência de Cauchy* se e só se

$$\forall \epsilon > 0 \exists l \in \mathbb{N} : j, k > l \Rightarrow \|x^{(k)} - x^{(j)}\| < \epsilon.$$

Generalizemos agora a noção de continuidade em \mathbb{R} a \mathbb{R}^n .

Definição 2.10 Uma função $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$ é *contínua em \hat{x}* se e só se

$$\|x - \hat{x}\| \rightarrow 0 \Rightarrow \|f(x) - f(\hat{x})\| \rightarrow 0.$$

Definição 2.11 Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função diferenciável. Define-se *derivada parcial em ordem a x_i no ponto x* como sendo

$$\frac{\partial f(x)}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(x + he_i) - f(x)}{h}.$$

Diz-se que f é de *classe C^n em D* , $f \in C^n(D)$, se qualquer que seja $x \in D$ as derivadas de ordem $1, 2, \dots, n$ são contínuas.

Define-se o *vector gradiente de f* como sendo

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right]^T$$

e a matriz Hessiana de f

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

São agora apresentados três casos particulares do *teorema de Taylor* sendo, principalmente os dois primeiros, bastante importantes para os métodos de otimização sem restrições.

Teorema 2.4 *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ com $f \in C^n(D)$, $n \geq 3$ e D um conjunto contendo o segmento de recta definido por x e $\hat{x} = x + h$, então existem $\theta_i \in (0, 1)$ ($i = 1, 2, 3$) tais que:*

- $f(\hat{x}) = f(x + h) = f(x) + \nabla f(\xi)^T h$ com $\xi = x + \theta_1 h$;
- $f(\hat{x}) = f(x + h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(\xi) h$ com $\xi = x + \theta_2 h$;
- $f(\hat{x}) = f(x + h) = f(x) + \nabla f(x)^T h + \frac{1}{2} h^T \nabla^2 f(x) h + R(x, \hat{x})$
com

$$R(x, \hat{x}) = \frac{1}{3!} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n h_i h_j h_k \frac{\partial^3 f(\xi)}{\partial x_i \partial x_j \partial x_k}$$

em que $h_r = \hat{x}_r - x_r$ e $\xi = x + \theta_3 h$.

Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função e seja D um subconjunto de \mathbb{R}^n .

Definição 2.12 O ponto x^* é um *minimizante local forte* de f sobre D se e só se existe $\epsilon > 0$ tal que:

1. $B(x^*, \epsilon) \subset D$;
2. $\forall x \in B(x^*, \epsilon) : x \neq x^* \Rightarrow f(x^*) < f(x)$.

Definição 2.13 O ponto x^* é um *minimizante local fraco* de f sobre D se e só se existe $\epsilon > 0$ tal que:

1. $B(x^*, \epsilon) \subset D$;
2. $\forall x \in B(x^*, \epsilon) : x \neq x^* \Rightarrow f(x^*) \leq f(x)$;
3. x^* não é minimizante local forte de F .

Note-se que uma função f pode ter mais do que um minimizante, sendo na prática difícil identificar um minimizante calculado numericamente como um minimizante global. Apenas se pode concluir que o ponto encontrado é minimizante numa dada vizinhança. Para determinar se um dado minimizante é global, é necessário conhecer todos os minimizantes e ver para qual deles a função objectivo tem valor mínimo. Em certos casos, pode não existir nenhum tipo de mínimo por:

1. f ser ilimitada ($f(x) = x^3$, $x \in \mathbb{R}$);
2. f ter mínimo no infinito de $\|x\|$ ($f(x) = e^{-x}$, $x \in \mathbb{R}$).

Definição 2.14 Seja $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ então $x^* \in D$ é um maximizante de f sobre D se e só se x^* for um minimizante de $-f$ sobre D .

As qualificações descritas atrás (local, global, fraco, forte) são obviamente também aplicáveis aos maximizantes.

Definição 2.15 Seja $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ uma função com primeiras derivadas parciais contínuas em D . $x^* \in D$ é um *ponto crítico* de f em D se e só se $\nabla f(x^*) = 0$. Caso x^* não seja nem minimizante nem maximizante então x^* diz-se um *ponto de sela*.

2.3 Síntese

Os resultados de Álgebra Linear e Análise que aqui foram introduzidos servem para melhor compreender as características dos problemas que irão ser tratados neste trabalho, bem como de alguns dos métodos descritos ao longo dos próximos três capítulos. Algumas definições são também utilizadas nas provas dos teoremas e propriedades.

Capítulo 3

Sistemas de Equações não Lineares

Neste capítulo são estudados métodos para a resolução de sistemas de equações não lineares. Como veremos mais adiante, este problema está relacionado com o problema de otimização.

Dado um vector de funções $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, com f continuamente diferenciável, pretende-se determinar $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ tal que $f(x^*) = 0$, ou seja, sendo $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ pretende-se encontrar uma solução para o sistema

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (3.1)$$

Os métodos estudados para a resolução deste problema são iterativos. Um dos mais conhecidos é o método de Newton.

No restante capítulo começamos por deduzir as equações do método de Newton e, em seguida, as do método Quasi-Newton do tipo de Broyden.

3.1 Método de Newton

Dado o sistema de equações (3.1), considere-se a expansão em *série de Taylor* de $f(x)$ em relação a $x = (x_1, x_2, \dots, x_n)$

$$\begin{cases} f_1(\bar{x}) = f_1(x) + (\bar{x}_1 - x_1) \frac{\partial f_1(x)}{\partial x_1} + \dots + (\bar{x}_n - x_n) \frac{\partial f_1(x)}{\partial x_n} + \dots \\ f_2(\bar{x}) = f_2(x) + (\bar{x}_1 - x_1) \frac{\partial f_2(x)}{\partial x_1} + \dots + (\bar{x}_n - x_n) \frac{\partial f_2(x)}{\partial x_n} + \dots \\ \vdots \\ f_n(\bar{x}) = f_n(x) + (\bar{x}_1 - x_1) \frac{\partial f_n(x)}{\partial x_1} + \dots + (\bar{x}_n - x_n) \frac{\partial f_n(x)}{\partial x_n} + \dots \end{cases} \quad (3.2)$$

com $\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$.

Assim, sendo

$$\bar{x} = x + d \quad (3.3)$$

e tomando apenas os $n + 1$ primeiros termos de cada equação, obtém-se a aproximação linear a f

$$l(\bar{x}) = f(x) + \nabla f(x)d, \quad (3.4)$$

onde a ∇f chamamos *Jacobiano de f* que é definido por

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}. \quad (3.5)$$

A solução do sistema linear $l(\bar{x}) = 0$ fornece uma aproximação à solução do sistema $f(\bar{x}) = 0$, e é dada por

$$\begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} d = - \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}. \quad (3.6)$$

Como $l(x)$ é apenas uma aproximação a $f(x)$, o vector \bar{x} é apenas uma aproximação à solução x^* do problema (3.1), pelo que o processo deve ser repetido, tomando como ponto de partida \bar{x} , $x \leftarrow \bar{x}$, e calculando uma nova aproximação a \bar{x} . Assim, por uma questão de notação, considerando que

partindo da aproximação $x^{(k)}$ se obtém $x^{(k+1)}$, usando (3.3) e (3.6), o processo iterativo do método de Newton, consiste em calcular

$$x^{(k+1)} = x^{(k)} + d^{(k)}, \quad k = 0, 1, 2, \dots$$

com

$$\nabla f(x^{(k)})d^{(k)} = -f(x^{(k)})$$

até que seja satisfeito o critério de paragem.

Algoritmo 3.1: [Método de Newton para a resolução do sistema de equações não lineares]

Dada a função f continuamente diferenciável e $x^{(0)} \in \mathbb{R}^n$

- Fazer $k := 0$
- **Repetir**
 - Determinar $d^{(k)}$ tal que $\nabla f(x^{(k)})d^{(k)} = -f(x^{(k)})$
 - Fazer $x^{(k+1)} := x^{(k)} + d^{(k)}$
 - Fazer $k := k + 1$

até convergir

- Fazer $x^* := x^{(k)}$

O teorema seguinte, estabelece as condições de convergência local quadrática do método de Newton

Teorema 3.1 *Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uma função continuamente diferenciável num conjunto aberto e convexo $D \subset \mathbb{R}^n$. Suponhamos que existe $x^* \in \mathbb{R}^n$ e $r, \beta \in \mathbb{R}^+$, tais que $B(x^*, r) \subset D$, $f(x^*) = 0$, $\nabla f(x^*)$ é não singular, $\|\nabla f(x^*)^{-1}\| \leq \beta$ e ∇f é Lipschitz contínua na vizinhança de x^* , em que a contante de Lipschitz é γ . Então existe $\epsilon \in \mathbb{R}^+$ tal que, qualquer que seja $x^{(0)} \in B(x^*, \epsilon)$ a sequência de pontos gerada pelo Algoritmo 3.1 converge para x^* e verifica*

$$\|x^{(k+1)} - x^*\| \leq \beta\gamma\|x^{(k)} - x^*\|^2, \quad k = 1, 2, 3, \dots$$

Demonstração: A demonstração deste resultado pode ser encontrada em [Fernandes, p.112] □

A razão de convergência do método de Newton é uma das suas vantagens. No entanto, o método não é globalmente convergente, necessitando de uma boa aproximação inicial.

A necessidade de calcular em cada iteração a solução do sistema (3.6), bem como o cálculo das derivadas da matriz (3.5), são os grandes inconvenientes que surgem na implementação do método. Outro problema poderá ainda vir da possibilidade da matriz do Jacobiano ser singular donde, o sistema (3.6) não é determinado.

O cálculo do Jacobiano (3.5), pode ser ultrapassado usando métodos que o aproximem, designadamente os métodos do tipo Quasi-Newton.

3.2 Método Quasi-Newton. Equação de Broyden

Quando o cálculo analítico das derivadas se torna difícil e/ou dispendioso, a matriz do Jacobiano deve ser aproximada por diferenças finitas ou actualizada usando fórmulas deduzidas a partir da equação secante.

Como já foi referido, seria vantajoso ter um processo que calculasse uma aproximação à matriz do Jacobiano, de modo que as derivadas não tivessem de ser recalculadas em cada iteração. Mais vantajoso ainda, seria determinar uma aproximação sem ter de realizar cálculos adicionais da função. Assim, nesta secção vamos introduzir a aproximação secante mais usada, neste contexto, proposta por Broyden.

Considere-se o modelo afim, que aproxima f em torno de x

$$l(x) = f(x^{(k+1)}) + A^{(k+1)}(x - x^{(k+1)}), \quad (3.7)$$

em que $l : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $A^{(k+1)} \in \mathbb{R}^{n \times n}$, $x = (x_1, x_2, \dots, x_n)^T$, $x^{(k+1)} = (x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)})^T$ e, como se verifica facilmente, temos $l(x^{(k+1)}) = f(x^{(k+1)})$ qualquer que seja a matriz $A^{(k+1)}$.

Ora, o método de Newton obtém-se substituindo, em cada iteração, $A^{(k+1)}$ por $\nabla f(x^{(k+1)})$. Quando o Jacobiano não é conhecido, a matriz $A^{(k+1)}$ deve então verificar

$$l(x^{(k)}) = f(x^{(k)}),$$

sendo $x^{(k)}$ o ponto da iteração corrente, k . Obtemos então

$$f(x^{(k)}) = l(x^{(k)}) = f(x^{(k+1)}) + A^{(k+1)}(x^{(k)} - x^{(k+1)})$$

donde

$$A^{(k+1)}(x^{(k+1)} - x^{(k)}) = f(x^{(k+1)}) - f(x^{(k)}).$$

Considerando $s^{(k)} = x^{(k+1)} - x^{(k)}$, obtemos

$$A^{(k+1)}s^{(k)} = y^{(k)}, \quad (3.8)$$

conhecida por *condição secante*, em que $s^{(k)}$ é o deslocamento efectuado e $y^{(k)} = f(x^{(k+1)}) - f(x^{(k)})$ a variação no resultado, obtida depois de efectuado o deslocamento.

A equação (3.8) não especifica um valor único para a matriz quadrada de ordem n , $A^{(k+1)}$. Apenas n elementos desta matriz podem ser determinados em função dos restantes $n \times (n - 1)$, que podem tomar quaisquer valores. Assim, para $s^{(k)} \neq 0$, existe um subespaço linear de matrizes de dimensão $n \times (n - 1)$ que satisfaz (3.8). A construção de uma boa aproximação secante, consiste em determinar um processo que seja bom para escolher uma dessas matrizes.

Por exemplo, podemos escolher $A^{(k+1)}$ por forma a minimizar a variação do modelo linear, quando passamos de $x^{(k)}$ para $x^{(k+1)}$ sujeito a (3.8). A diferença entre o novo modelo, $l(x) = f(x^{(k+1)}) + A^{(k+1)}(x - x^{(k+1)})$ dado em (3.7), e o modelo antigo, $l^{(k)}(x) = f(x^{(k)}) + A^{(k)}(x - x^{(k)})$ é dada por

$$\begin{aligned} l(x) - l^{(k)}(x) &= f(x^{(k+1)}) + A^{(k+1)}(x - x^{(k+1)}) - \\ &\quad - f(x^{(k)}) - A^{(k)}(x - x^{(k)}) \\ &= f(x^{(k+1)}) - f(x^{(k)}) - A^{(k+1)}(x^{(k+1)} - x^{(k)}) + \\ &\quad + (A^{(k+1)} - A^{(k)})(x - x^{(k)}) \end{aligned}$$

e agora, tendo em conta a equação secante (3.8), vem

$$l(x) - l^{(k)}(x) = (A^{(k+1)} - A^{(k)})(x - x^{(k)}). \quad (3.9)$$

Podemos agora escrever $x - x^{(k)}$ como soma de um múltiplo do vector $s^{(k)}$ com um vector, t , que é ortogonal a $s^{(k)}$

$$x - x^{(k)} = \alpha s^{(k)} + t = \alpha(x^{(k+1)} - x^{(k)}) + t$$

donde, podemos escrever (3.9) da seguinte forma

$$l(x) - l^{(k)}(x) = \alpha (A^{(k+1)} - A^{(k)}) (x^{(k+1)} - x^{(k)}) + (A^{(k+1)} - A^{(k)}) t. \quad (3.10)$$

Não tendo qualquer controlo sobre o primeiro termo do segundo membro da equação (3.10) pois, a condição secante implica que

$$\begin{aligned} (A^{(k+1)} - A^{(k)}) (x^{(k+1)} - x^{(k)}) &= A^{(k+1)}(x^{(k+1)} - x^{(k)}) - \\ &\quad - A^{(k)}(x^{(k+1)} - x^{(k)}) = \\ &= y^{(k)} - A^{(k)}s^{(k)}, \end{aligned}$$

podemos escolher $A^{(k+1)}$ por forma que

$$(A^{(k+1)} - A^{(k)})t = 0,$$

tendo em conta que t e $s^{(k)}$ são ortogonais. Por este facto, $A^{(k+1)} - A^{(k)}$ deverá ser uma matriz de característica um da forma $us^{(k)T}$, $u \in \mathbb{R}^n$. Ou seja, uma matriz cujas linhas são vectores múltiplos de $s^{(k)}$ o que implica que $us^{(k)T}t = 0$. Subtraindo a ambos os membros da condição secante, (3.8), o termo $A^{(k)}s^{(k)}$, obtemos

$$(A^{(k+1)} - A^{(k)})s^{(k)} = y^{(k)} - A^{(k)}s^{(k)}.$$

Tendo em conta que estamos a considerar

$$A^{(k+1)} - A^{(k)} = us^{(k)T}, \quad (3.11)$$

vem

$$us^{(k)T}s^{(k)} = y^{(k)} - A^{(k)}s^{(k)}$$

donde

$$u = \frac{y^{(k)} - A^{(k)}s^{(k)}}{s^{(k)T}s^{(k)}}. \quad (3.12)$$

Assim, de (3.11) e (3.12) obtemos

$$A^{(k+1)} = A^{(k)} + \frac{(y^{(k)} - A^{(k)}s^{(k)})s^{(k)T}}{s^{(k)T}s^{(k)}}. \quad (3.13)$$

denominada *fórmula de actualização de Broyden*.

No processo iterativo é necessário possuir uma matriz inicial, $A^{(0)}$, podendo considerar-se, por exemplo, uma das três hipóteses seguintes:

Aproximação inicial $A^{(0)}$	Observações
$\nabla f(x^{(0)})$	Necessário o cálculo analítico das derivadas da função
A matriz que se obtém aproximando as derivadas do Jacobiano, usando <i>diferenças finitas</i> , calculada no ponto $x^{(0)}$	Necessário realizar muitos cálculos da função f
A matriz identidade	Implementação muito simples

Podemos agora construir um algoritmo que utiliza a fórmula de actualização de Broyden para aproximar a matriz do Jacobiano.

Algoritmo 3.2: [Método de Quasi-Newton-Broyden para a resolução do sistema de equações não lineares]

Dada a função f continuamente diferenciável, $x^{(0)} \in \mathbb{R}^n$ e, por exemplo, $A^{(0)} = I_n$

- Fazer $k := 0$
- **Repetir**
 - Determinar $s^{(k)}$ tal que $A^{(k)}s^{(k)} = -f(x^{(k)})$
 - Fazer $x^{(k+1)} := x^{(k)} + s^{(k)}$
 - **Se não convergiu**
 - * Fazer $y^{(k)} := f(x^{(k+1)}) - f(x^{(k)})$
 - * Determinar $A^{(k+1)} := A^{(k)} + \frac{(y^{(k)} - A^{(k)}s^{(k)})s^{(k)T}}{s^{(k)T}s^{(k)}}$
 - * Fazer $k := k + 1$

até convergir

- Fazer $x^* := x^{(k+1)}$

Como seria de esperar, o comportamento deste método vai depender tanto da aproximação inicial à solução, $x^{(0)}$, como da aproximação inicial à matriz do Jacobiano. Para evitar a divergência de $A^{(k)}$, será útil implementar uma técnica de recomeço, que consiste em repôr em $A^{(k)}$ a matriz usada como $A^{(0)}$ de n em n iterações.

Os teoremas que se seguem referem que o método tem convergência local, isto é, dado um ponto inicial, $x^{(0)}$, suficientemente próximo da solução x^* e

uma aproximação inicial, $A^{(0)}$, também suficientemente próxima da matriz $\nabla f(x^*)$, a convergência é superlinear para x^* .

Teorema 3.2 *Seja $D \in \mathbb{R}^n$ um conjunto aberto e convexo que contém $x^{(k)}$ e $x^{(k+1)}$, com $x^{(k)} \neq x^*$. Considere-se ainda a função $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, a matriz do Jacobiano ∇f Lipschitz contínua em D , com constante γ , $A^{(k)} \in \mathbb{R}^{n \times n}$ e $A^{(k+1)}$ determinada por (3.13). Então, quer para a norma 2 quer para a norma de Frobenius, tem-se*

$$\|A^{(k+1)} - \nabla f(x^{(k+1)})\| \leq \|A^{(k)} - \nabla f(x^{(k)})\| + \frac{3\gamma}{2} \|x^{(k+1)} - x^{(k)}\|_2. \quad (3.14)$$

Além disso, se $x^* \in D$ e ∇f satisfaz a condição de Lipschitz fraca

$$\|\nabla f(x) - \nabla f(x^*)\| \leq \gamma \|x - x^*\|, \forall x \in D$$

então

$$\begin{aligned} \|A^{(k+1)} - \nabla f(x^*)\| &\leq \|A^{(k)} - \nabla f(x^*)\| + \\ &+ \frac{\gamma}{2} (\|x^{(k+1)} - x^*\|_2 + \|x^{(k)} - x^*\|_2). \end{aligned} \quad (3.15)$$

Demonstração: A demonstração deste resultado pode ser encontrada em [Dennis-Schnabel, p.176] \square

As desigualdades (3.14) e (3.15) são exemplos da chamada *propriedade da deterioração limitada*. Esta propriedade indica que se a aproximação à matriz do Jacobiano piorar, de iteração para iteração, esta piora será de uma forma controlada e limitada.

Teorema 3.3 *Suponhamos que se verificam as hipóteses do Teorema 3.2. Suponhamos ainda que existem escalares positivos ϵ e δ tais que $\|x^{(0)} - x^*\|_2 \leq \epsilon$ e $\|A^{(0)} - \nabla f(x^*)\| \leq \delta$. Então a sequência $\{x^{(k)}\}$ gerada pelo Algoritmo 3.2 converge superlinearmente para x^* . Caso $\{A^{(k)}\}$ só satisfaça a desigualdade (3.15) então a convergência é pelo menos linear para x^* .*

Demonstração: A demonstração deste resultado pode ser encontrada em [Dennis-Schnabel, p.177] \square

A propriedade da deterioração limitada assegura que $\|A^{(k+1)} - \nabla f(x^*)\|$ se conserva razoavelmente pequena. Se as aproximações ao Jacobiano apenas satisfazem (3.15), então não se consegue mais do que uma convergência linear. Isto significa que o processo iterativo converge, mesmo que a aproximação ao Jacobiano não seja muito boa. No entanto, o que nos interessa é ter convergência superlinear. Esta consegue-se localmente de acordo com as condições do Teorema 3.3.

Uma condição necessária e suficiente para que um método do tipo secante, e nomeadamente o método de Broyden, convirja superlinearmente para x^* é a de que os deslocamentos efectuados pelo método secante convirjam, quer em direcção quer em amplitude para os deslocamentos Newton, quando efectuados a partir dos mesmos pontos. O teorema seguinte estabelece isto mesmo.

Teorema 3.4 *Seja $D \subset \mathbb{R}^n$ um conjunto aberto e convexo, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\nabla f(x)$ Lipschitz contínua em D com constante γ e $\nabla f(x^*)$ não singular. Seja $\{A^{(k)}\}$ uma sequência de matrizes não singulares em $\mathbb{R}^{n \times n}$ e suponha-se que para um $x^{(0)} \in D$ a sequência de pontos gerada por*

$$x^{(k+1)} = x^{(k)} + s^{(k)},$$

com $A^{(k)}s^{(k)} = -f(x^{(k)})$ permanece no conjunto D e satisfaz $x^{(k)} \neq x^$, para qualquer k , e*

$$\lim_{k \rightarrow \infty} x^{(k)} = x^*.$$

Então a sequência $\{x^{(k)}\}$ converge superlinearmente para x^ , de acordo com alguma norma, e $f(x^*) = 0$ se e só se*

$$\lim_{k \rightarrow \infty} \frac{\|(A^{(k)} - \nabla f(x^*))s^{(k)}\|}{\|s^{(k)}\|} = 0. \quad (3.16)$$

Demonstração: A demonstração deste resultado pode ser encontrada em [Dennis-Schnabel, p.181] □

De acordo com a equação (3.16), quando o método Quasi-Newton converge superlinearmente para x^* , não é possível concluir que a matriz $A^{(k)}$, que aproxima o Jacobiano, esteja a convergir para $\nabla f(x^*)$, embora possa acontecer.

3.3 Síntese

Neste capítulo foi introduzida a formulação do problema definido por um sistema de n equações não lineares nas n variáveis reais e foram deduzidas as equações iterativas de dois dos métodos mais usados para a sua resolução: o método de Newton e um método do tipo secante, conhecido na literatura por Quasi-Newton, e que é baseado na fórmula de actualização de Broyden para a matriz do Jacobiano do sistema.

Este problema tem uma forte ligação com o problema que será tratado no próximo capítulo. Além disso, no capítulo 6 recorre-se à fórmula de actualização de Broyden para aproximar uma matriz do Jacobiano, na sequência do problema tratado no capítulo 5.

Capítulo 4

Optimização Sem Restrições

Como se sabe, alguns problemas de otimização envolvem a minimização de uma função, possivelmente não linear, podendo as suas variáveis ter de satisfazer um conjunto de restrições. Quando não existem restrições, temos um problema de otimização não linear sem restrições, que pode ser expresso pela seguinte forma matemática:

$$\min_{x \in \mathbb{R}^n} f(x). \quad (4.1)$$

Quando f é continuamente diferenciável até à segunda ordem, existe um conjunto de métodos para a resolução destes problemas, sendo o método de Newton o mais conhecido. Assim, neste capítulo vamos começar por definir as condições de optimalidade e em seguida serão deduzidas as equações do método de Newton. Por fim veremos algumas modificações que têm como objectivo conseguir obter convergência global e tentar reduzir os custos computacionais.

4.1 Condições de Optimalidade

Antes de começarmos a pensar em métodos para a resolução dos problemas do tipo dos definidos pela expressão (4.1), temos de conseguir definir uma "solução" do problema, ou seja, as condições de optimalidade. Estas condições vão permitir verificar se um dado ponto é óptimo e, como veremos, vão ainda sugerir métodos para a resolução do próprio problema.

Ora, nas Definições 2.12 e 2.13 foram indicadas as condições de mínimo local forte e fraco, respectivamente. No entanto, estas definições não

servem para verificar se um dado ponto é ou não óptimo pois, em geral, seria necessário calcular f no conjunto (infinito) de todos os pontos de uma ϵ -vizinhança do ponto proposto para verificação. Começemos então por definir as chamadas condições necessárias, para que um ponto x^* seja mínimo local do problema (4.1)[Gill–Murray–Wright, p.63].

Teorema 4.1 *Suponhamos que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ tem derivadas contínuas até à segunda ordem num conjunto convexo D . As condições necessárias para que x^* seja um mínimo local de f são:*

CN1. $\nabla f(x^*) = 0$ (ou equivalentemente $\|\nabla f(x^*)\| = 0$).

CN2. $\nabla^2 f(x^*) \in PSD$.

Estas condições são chamadas condições necessárias de primeira e segunda ordem, respectivamente.

Demonstração: Como sabemos, o desenvolvimento em série de Taylor de f em torno do ponto x^* é dado por

$$f(x^* + \epsilon d) = f(x^*) + \epsilon d^T \nabla f(x^*) + \frac{1}{2} \epsilon^2 d^T \nabla^2 f(x^* + \epsilon \theta d) d, \quad (4.2)$$

com $0 \leq \theta \leq 1$, $d \in \mathbb{R}^n$ e considerando ainda (sem perda de generalidade) $\epsilon \in \mathbb{R}^+$.

Suponhamos que x^* é um mínimo local, não sendo, no entanto, um ponto crítico, isto é, $\nabla f(x^*) \neq 0$. Assim, existe pelo menos um vector $d \in \mathbb{R}^n$ tal que

$$d^T \nabla f(x^*) < 0 \quad (4.3)$$

(basta, por exemplo, considerar $d = -\nabla f(x^*)$). Qualquer vector d que satisfaça (4.3) diz-se uma *direcção descendente*. Dada uma direcção descendente d , existe um escalar $\bar{\epsilon}$ ($\bar{\epsilon} > 0$) tal que para todos os ϵ ($0 < \epsilon \leq \bar{\epsilon}$), e tendo em conta que quando $\epsilon \rightarrow 0$ então $\epsilon^2 \rightarrow 0$ mais rapidamente, vem

$$\epsilon d^T \nabla f(x^*) + \frac{1}{2} \epsilon^2 d^T \nabla^2 f(x^* + \epsilon \theta d) d < 0. \quad (4.4)$$

Consequentemente, de (4.2) e (4.4), concluímos que

$$f(x^* + \epsilon d) < f(x^*),$$

qualquer que seja o $\epsilon \leq \bar{\epsilon}$, o que contradiz a nossa hipótese de x^* ser mínimo. Conclui-se então que, exceptuando quando $\nabla f(x^*) = 0$, toda a vizinhança de x^* contém pontos com um valor estritamente menor do que $f(x^*)$, o que prova que qualquer mínimo local tem ser um ponto crítico.

A necessidade da condição de segunda ordem, pode também ser provada por contradição. Note-se que de (4.2) e da condição de primeira ordem já provada, se tem

$$f(x^* + \epsilon d) = f(x^*) + \frac{1}{2}\epsilon^2 d^T \nabla^2 f(x^* + \epsilon \theta d) d. \quad (4.5)$$

Supondo que a matriz $\nabla^2 f(x^*)$ é indefinida, pela continuidade de $\nabla^2 f$, esta será indefinida numa vizinhança de x^* , pelo que se pode escolher um $\epsilon > 0$ em (4.5), suficientemente pequeno, por forma que $x^* + \epsilon d$ pertença à região referida. Por definição de matriz indefinida, podemos escolher d por forma que $d^T \nabla^2 f(x^* + \epsilon \theta d) d < 0$. Assim, de (4.5), qualquer vizinhança de x^* contém pontos onde f tem valores estritamente inferiores a $f(x^*)$, o que contradiz a optimalidade de x^* .

□

As condições para verificar que um ponto é óptimo são:

Teorema 4.2 *Suponhamos que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ tem derivadas contínuas até à segunda ordem num conjunto convexo $D \subseteq \mathbb{R}^n$. As condições suficientes para que x^* seja um mínimo local forte de f são:*

CS1. $\nabla f(x^*) = 0$ (ou equivalentemente $\|\nabla f(x^*)\| = 0$).

CS2. $\nabla^2 f(x^*) \in PD$.

Estas condições são chamadas condições suficientes de primeira e segunda ordem, respectivamente.

Demonstração: Começemos por ter em conta que já foi verificado a necessidade de $\nabla f(x^*) = 0$, pelo que a expansão de f em torno de x^* é dada por (4.5). Supondo $\nabla^2 f \in PD$, por continuidade, $\nabla^2 f$ é positiva definida numa vizinhança de x^* . Para $|\epsilon|$ suficientemente pequeno, temos que $x^* + \epsilon d$ pertence à referida vizinhança. Assim,

para todos os ϵ nas referidas condições e para qualquer direcção d verifica-se $d^T \nabla^2 f(x^* + \epsilon \theta d) d > 0$. Recorrendo novamente a (4.5) concluímos que $f(x^*)$ é estritamente inferior a qualquer ponto na referida vizinhança, pelo que x^* é um mínimo local forte. \square

4.2 Método de Newton

Nesta secção vamos deduzir o método de Newton, tratando-se de um método de segunda ordem pois, como veremos, recorre à Hessiana de f .

Começemos por considerar a expansão em série de Taylor da função objectivo, f , em torno de um ponto x , dada por

$$f(x^*) \approx f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d. \quad (4.6)$$

Relembrando o objectivo de resolver o problema (4.1), o ideal seria que a partir de qualquer ponto x se conseguisse determinar o vector d , por forma a atingir o mínimo, $x^* = x + d$. Ora, $f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d$ define uma função quadrática, que aproxima f na vizinhança de x e cujo mínimo é d . Para determinar d , fixa-se o x , e considera-se a partir de (4.6) a aplicação

$$\phi(d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d. \quad (4.7)$$

Derivando ambos os membros de (4.7) em ordem a d , obtemos

$$\nabla_d \phi(d) = \nabla f(x) + \nabla^2 f(x) d.$$

Para que d seja mínimo de ϕ , terá de ser ponto crítico, ou seja,

$$\nabla f(x) + \nabla^2 f(x) d = 0,$$

que de forma matricial fica

$$\nabla^2 f(x) d = -\nabla f(x). \quad (4.8)$$

A (4.8) chamamos *sistema Newton*. Como as derivadas de segunda ordem das funções quadráticas são constantes, o mínimo de (4.7) é atingido numa só etapa, a partir de um ponto inicial qualquer x . Quando f não é uma função

quadrática o ponto $x + d$, com d obtido de (4.8) não será necessariamente o mínimo, pelo que o processo deverá ser repetido, partindo de $x + d$.

Uma abordagem diferente, recorre ao problema descrito no Capítulo 3. Assim, dada $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$, em que f possui derivadas contínuas pelo menos até à segunda ordem, D é um conjunto convexo que contém pelo menos um ponto crítico, x^* , de f e $\nabla^2 f(x^*)$ é definida positiva, então x^* é um minimizante local forte de f em D .

Assim, um método que resolva o sistema de equações não lineares

$$\nabla f(x) = 0 \equiv \begin{cases} \frac{\partial f(x)}{\partial x_1} = 0 \\ \frac{\partial f(x)}{\partial x_2} = 0 \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} = 0 \end{cases} \quad (4.9)$$

serve para determinar x^* .

Um método eficiente para a resolução do sistema de equações não lineares (4.9) é o método de Newton (ver Capítulo 3).

Por qualquer das abordagens, obtemos um processo iterativo, sendo

$$x^{(k+1)} = x^{(k)} + d^{(k)}, \quad (k = 0, 1, 2, \dots)$$

com

$$\nabla^2 f(x^{(k)})d^{(k)} = -\nabla f(x^{(k)}). \quad (4.10)$$

Algoritmo 4.1: [Método de Newton para minimização de funções não lineares sem restrições]

Dada a função f continuamente diferenciável pelo menos até à segunda ordem e $x^{(0)} \in \mathbb{R}^n$

- Fazer $k := 0$
- **Repetir**
 - Determinar $d^{(k)}$ tal que $\nabla^2 f(x^{(k)})d^{(k)} = -\nabla f(x^{(k)})$
 - Fazer $x^{(k+1)} := x^{(k)} + d^{(k)}$
 - Fazer $k := k + 1$

até convergir

- Fazer $x^* := x^{(k)}$

Uma das propriedades mais importantes do método de Newton, a sua convergência quadrática, é dada no seguinte teorema.

Teorema 4.3 *Suponhamos que $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ tem derivadas parciais contínuas num conjunto aberto e convexo D . Supondo ainda que existem $x^* \in \mathbb{R}^n$ e $r, \beta > 0$, tais que*

- $B(x^*, r) \subset D$
- $\nabla f(x^*) = 0$
- existe $\nabla^2 f(x^*)^{-1}$ e $\|\nabla^2 f(x^*)^{-1}\| < \beta$ (isto é, $\nabla^2 f(x^*)^{-1}$ é limitada)
- $\nabla^2 f(x^*)$ é γ -Lipschitz contínua em $B(x^*, r)$

então existe um $\epsilon > 0$ tal que qualquer que seja $x \in B(x^*, \epsilon)$ a sequência $x^{(1)}, x^{(2)}, \dots$ gerada por

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

fica bem definida, converge para x^* e satisfaz

$$\|x^{(k+1)} - x^*\| \leq \beta\gamma \|x^{(k)} - x^*\|^2, \quad k = 0, 1, 2, \dots \quad (4.11)$$

Demonstração: A demonstração deste resultado pode ser encontrada em [Dennis-Schnabel, pag.90]

□

Assim, pelo Teorema 4.3 que acabámos de enunciar, concluí-se que se a aproximação inicial, $x^{(0)}$, estiver suficientemente próxima de x^* , então a sequência de pontos gerada pelo método de Newton converge para x^* e a convergência é quadrática. Contudo, se a aproximação inicial não for suficientemente boa, o método pode não convergir. Existem ainda outras razões que levam o método a falhar a convergência, nomeadamente:

- O vector $d^{(k)}$, calculado de (4.10), é quase ortogonal a $\nabla f(x^{(k)})$ pelo que não é possível progredir ao longo de $d^{(k)}$.
- Quando $d^{(k)}$ não é uma direcção descendente não se pode garantir a existência de um valor α tal que $f(x^{(k)} + \alpha d^{(k)}) < f(x^{(k)})$.
- Se a matriz Hessiana calculada em $x^{(k)}$ é singular então a direcção $d^{(k)}$ não está sequer definida.

- Apesar de $\nabla^2 f(x^{(k)})^{-1}$ existir e ser positiva definida e a direcção, $d^{(k)}$, calculada de (4.10) ser descendente, temos $f(x^{(k+1)}) > f(x^{(k)})$, isto é, a função não sofreu uma redução.

Atendendo a estas considerações, podem realizar-se algumas modificações no método de Newton por forma a torná-lo mais robusto.

Uma primeira modificação, tem a ver com a primeira razão referida. Assim, quando a direcção calculada, $d^{(k)}$, for quase ortogonal a $\nabla f(x^{(k)})$, esta ($d^{(k)}$) deve ser substituída por outra, nomeadamente pela direcção de descida máxima¹, $-\nabla f(x^{(k)})$. Na prática, devido aos arredondamentos, vamos considerar que $d^{(k)}$ e $\nabla f(x^{(k)})$ são ortogonais quando

$$|\nabla f(x^{(k)})^T d^{(k)}| \leq \eta \|\nabla f(x^{(k)})\| \|d^{(k)}\|,$$

com $\eta > 0$ e próximo de zero.

Uma segunda modificação tem como objectivo resolver o problema indicado na segunda observação. Assim, caso $d^{(k)}$ não seja uma direcção descendente devemos considerar a nossa direcção como sendo $-d^{(k)}$.

Quando a matriz Hessiana for singular, terceira observação, como a direcção não pode ser calculada, toma-se $d^{(k)} = -\nabla f(x^{(k)})$.

Finalmente, tendo em conta a última observação, e analogamente ao que foi feito na demonstração do Teorema 4.1, podemos garantir a existência, em cada iteração, de um escalar $\alpha^{(k)}$ denominado de *comprimento do passo*, tal que

$$f(x^{(k)} + \alpha^{(k)} d^{(k)}) < f(x^{(k)}).$$

O cálculo desse escalar $\alpha^{(k)}$, é realizado usando uma procura unidimensional, segundo a direcção de procura $d^{(k)}$. Ou seja, pretende-se determinar

$$\min_{\alpha \in \mathbb{R}} f(x^{(k)} + \alpha d^{(k)}).$$

Para esta procura pode-se, por exemplo, recorrer:

- A uma procura exacta, obtendo-se o mínimo local de f segundo a direcção $d^{(k)}$;
- A uma aproximação desse mínimo através de um método de minimização unidimensional (por exemplo, o método de *Davies, Swann e Campey (DSC)*[Abdy-Dempster, p.36], baseado em aproximações quadráticas ou cúbica de f segundo a direcção d);

¹ou descida abrupta (tradução de *steepest descent*)

- Ao *critério de Armijo* que origina uma redução significativa do valor de f . Neste caso, pressupondo que a direcção $d^{(k)}$ é descendente, gera-se uma sequência de escalares $\{w^j\alpha\}$ ($j = 0, 1, 2, \dots$), a partir de um valor inicial α (sendo a unidade o valor mais comum) e sendo $0 < w < 1$ uma constante de redução. Quando se encontrar o primeiro escalar w^j que verifique

$$f(x^{(k)}) - f(x^{(k)} + w^j\alpha d^{(k)}) \geq \mu_1 w^j \alpha \nabla f(x^{(k)})^T d^{(k)}, \quad (4.12)$$

faz-se $\alpha^{(k)} = w^j\alpha$. No entanto, valores muito grandes, não são aceites por esta *condição de Armijo*, embora possam passar valores muito pequenos. Para ultrapassar isto, é conveniente juntar a condição adicional

$$f(x^{(k)}) - f(x^{(k)} + w^j\alpha d^{(k)}) \leq \mu_2 w^j \alpha \nabla f(x^{(k)})^T d^{(k)}. \quad (4.13)$$

Ao conjunto das duas condições chamamos *critério de Goldstein-Armijo*. As constantes μ_1 e μ_2 deverão satisfazer $0 < \mu_1 \leq \mu_2 < 1$. Sendo que quanto menor for μ_1 menor é a redução verificada no valor da função, em cada iteração, mais facilmente é verificada a condição (4.12) e maior é o número de iterações pois, o processo progride mais lentamente. Por sua vez quanto maior for μ_1 , mais difícil é de satisfazer a condição (4.12) e maior é a redução verificada no valor da função objectivo pelo que são necessárias, em geral, menos iterações. Pode ainda acontecer que o valor de $\alpha^{(k)}$ aceitável, que verifica as condições (4.12) e (4.13) seja muito pequeno e consequentemente o deslocamento resultante de $\alpha^{(k)}d^{(k)}$ é quase nulo. Esta situação, que ocorre quando já se está muito próximo do óptimo, obriga a um número exagerado de cálculos da função objectivo, pelo que se sugere utilizar $\alpha^{(k)} = 1$ (mesmo que não exista decréscimo do valor da função).

Por vezes, os métodos mais simples e menos precisos são preferidos, uma vez que uma redução, considerada significativa, no valor da função é suficiente para garantir convergência[Fletcher, p.30].

A introdução destes esquemas de aviso das falhas do método de Newton, a implementação das soluções descritas bem como a introdução da pesquisa unidimensional, origina o chamado *algoritmo de segurança de Newton* [Wolfe, p.92].

Algoritmo 4.2: [Algoritmo de Segurança de Newton para minimização de funções não lineares sem restrições]

Dada a função f continuamente diferenciável pelo menos até à segunda ordem e $x^{(0)} \in \mathbb{R}^n$

- Fazer $k := 0$

- **Repetir**

- Determinar, se possível, $d^{(k)}$ solução do sistema

$$\nabla^2 f(x^{(k)})d^{(k)} = -\nabla f(x^{(k)})$$

- Se o sistema tem solução única

- então**

- * Se $|\nabla f(x^{(k)})^T d^{(k)}| \leq \eta \|\nabla f(x^{(k)})\|_2 \|d^{(k)}\|_2$

- então** $d^{(k)} := -\nabla f(x^{(k)})$

- senão**

- Se $\nabla f(x^{(k)})^T d^{(k)} > \eta \|\nabla f(x^{(k)})\|_2 \|d^{(k)}\|_2$

- então** $d^{(k)} := -d^{(k)}$

- senão** $d^{(k)} := -\nabla f(x^{(k)})$

- Determinar $\alpha^{(k)}$ usando uma pesquisa unidimensional

- Fazer $x^{(k+1)} := x^{(k)} + \alpha^{(k)} d^{(k)}$

- Fazer $k := k + 1$

- até convergir**

- Fazer $x^* := x^{(k)}$

4.3 Métodos Quasi-Newton

Como vimos na secção anterior, quando for dada uma aproximação inicial suficientemente boa, o método de Newton é relativamente rápido. Contudo, é usual ter de se recorrer a um elevado esforço computacional ao longo do processo iterativo. Esse esforço computacional requiere em cada iteração:

- o cálculo analítico das derivadas da Hessiana da função objectivo, o que corresponde a determinar $n(n+1)/2$ derivadas parciais de segunda

ordem².

- O cálculo da matriz $(\nabla^2 f(x^{(k)}))^{-1}$, ou equivalentemente, a resolução de um sistema linear. Qualquer dos casos requer operações aritméticas da ordem de $\mathcal{O}(n^3)$.

Estes inconvenientes, tornam o método de Newton pouco indicado para implementação em certos casos.

Todas estas considerações levam a que se tente usar métodos, para minimizar f , que não necessitem de determinar analiticamente as segundas derivadas, nem de resolver sistemas. Assim, o objectivo é aproximar $\nabla^2 f(x^{(k)})$ ou $(\nabla^2 f(x^{(k)}))^{-1}$ usando informação disponível baseada em $x^{(k)}$, $\nabla f(x^{(k)})$, $x^{(k-1)}$ e $\nabla f(x^{(k-1)})$. No caso em que se aproxima a Hessiana,

$$B^{(k)} \approx \nabla^2 f(x^{(k)}), \quad (4.14)$$

o ponto seguinte do processo iterativo é

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} (B^{(k)})^{-1} \nabla f(x^{(k)}).$$

No caso em que se aproxima, desde logo, a inversa da Hessiana

$$H^{(k)} \approx (\nabla^2 f(x^{(k)}))^{-1} \quad (4.15)$$

o cálculo do ponto seguinte do processo iterativo é obtido de

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} H^{(k)} \nabla f(x^{(k)}),$$

com $\alpha^{(k)}$ escalar determinado na procura unidimensional.

A teoria dos métodos Quasi-Newton é pois baseada no facto de se poder calcular uma aproximação à curvatura da função objectivo não linear, sem que para tal seja necessário determinar explicitamente a matriz Hessiana da função.

Supondo que f possui derivadas contínuas até à segunda ordem, a expansão em série de Taylor do gradiente de f em torno de $x^{(k)}$ é:

$$\nabla f(x^{(k+1)}) = \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)})(x^{(k+1)} - x^{(k)}) + \Delta^{(k)}, \quad (4.16)$$

²Este problema poderia até certo ponto ser resolvido usando diferenças finitas para estimar $\nabla^2 f(x^{(k)})$. No entanto, ao fazer $\frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_i \partial x_j} \approx \frac{1}{h} \left(\frac{\partial f(x_1, \dots, x_j+h, \dots, x_n)}{\partial x_i} - \frac{\partial f(x_1, \dots, x_n)}{\partial x_i} \right)$, é necessário determinar $\frac{\partial f(x_1, \dots, x_j+h, \dots, x_n)}{\partial x_i}$, com $1 \leq i \leq j \leq n$, num total de $n(n+1)/2$ cálculos. Por estes motivos, o recurso a diferenças finitas pode ser pouco eficiente para n grande ou quando o custo computacional de determinar ∇f for elevado.

com $\Delta^{(k)} \rightarrow 0$ quando $x^{(k+1)} \rightarrow x^{(k)}$. Considerando

$$s^{(k)} = x^{(k+1)} - x^{(k)},$$

$$y^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$$

e truncando o termo $\Delta^{(k)}$, vem de (4.16)

$$y^{(k)} \approx \nabla^2 f(x^{(k)}) s^{(k)}.$$

Aproximando $\nabla^2 f(x^{(k)})$ por $B^{(k)}$, obtemos

$$y^{(k)} = B^{(k)} s^{(k)} \quad (k \geq 0)$$

e a correspondente equação para a aproximação à inversa da Hessiana, $H^{(k)}$, é

$$H^{(k)} y^{(k)} = s^{(k)}, \quad (k \geq 0).$$

Por razões de sequenciação (cálculo), estas duas equações terão de ser substituídas, respectivamente, por

$$y^{(k)} = B^{(k+1)} s^{(k)}, \quad (k \geq 0) \tag{4.17}$$

e

$$H^{(k+1)} y^{(k)} = s^{(k)}, \quad (k \geq 0). \tag{4.18}$$

A equação (4.17) é conhecida por *condição Quasi-Newton ou secante*.

Agora, seria desejável poder calcular $B^{(k+1)}$, se possível adicionando a $B^{(k)}$ uma matriz de correcção, $C^{(k)}$, que dependesse de $B^{(k)}$, $s^{(k)}$ e $y^{(k)}$, ou seja,

$$B^{(k+1)} = B^{(k)} + C^{(k)} \left(B^{(k)}, s^{(k)}, y^{(k)} \right), \quad k = 0, 1, 2, \dots$$

De modo análogo,

$$H^{(k+1)} = H^{(k)} + E^{(k)} \left(H^{(k)}, s^{(k)}, y^{(k)} \right), \quad k = 0, 1, 2, \dots,$$

tomando $E^{(k)}$ como matriz de correcção.

4.3.1 Fórmulas de Actualização Para a Hessiana

Durante uma iteração, novas informações acerca da curvatura da função objectivo, f , são obtidas ao longo de uma única direcção, pelo que será de esperar que $B^{(k+1)}$ difira de $B^{(k)}$ por uma matriz de característica inferior a n . De facto, a condição Quasi-Newton pode ser satisfeita adicionando uma matriz com característica um, dois ou matrizes de norma mínima. Considerando o caso de uma matriz de característica um, suponhamos

$$B^{(k+1)} = B^{(k)} + uv^T \quad (4.19)$$

para determinados vectores u e v . Da condição Quasi-Newton (4.17) temos:

$$B^{(k+1)}s^{(k)} = (B^{(k)} + uv^T)s^{(k)} = y^{(k)},$$

ou de uma forma equivalente,

$$u(v^T s^{(k)}) = y^{(k)} - B^{(k)}s^{(k)}.$$

Agora, pelo facto de $v^T s^{(k)}$ ser um escalar e assumindo que $y^{(k)}$ é diferente de $B^{(k)}s^{(k)}$ (caso contrário, $B^{(k)}$ já satisfazia a condição Quasi-Newton) concluimos que u tem a mesma direcção de $y^{(k)} - B^{(k)}s^{(k)}$. Assim, para qualquer vector v tal que $v^T s^{(k)} \neq 0$ vem:

$$u = \frac{y^{(k)} - B^{(k)}s^{(k)}}{v^T s^{(k)}}.$$

Consequentemente, $B^{(k+1)}$ fica definida por

$$B^{(k+1)} = B^{(k)} + \frac{1}{v^T s^{(k)}} \left(y^{(k)} - B^{(k)}s^{(k)} \right) v^T. \quad (4.20)$$

Dado um vector w ortogonal a $s^{(k)}$, o produto da matriz de característica um zw^T por $s^{(k)}$, $(zw^T)s^{(k)}$, é o vector nulo. Assim, a condição Quasi-Newton (4.17) continua a ser satisfeita se forem adicionadas uma ou mais matrizes de característica um e da forma zw^T , sendo contudo os elementos de $B^{(k+1)}$ alterados por cada nova matriz adicionada. Como a condição Quasi-Newton por si só não determina de forma única a matriz de correcção $C^{(k)}$, será útil colocar certas condições por forma a que $B^{(k+1)}$ satisfaça algumas propriedades desejáveis, nomeadamente a simetria e que seja positiva definida.

Simetria

Visto a matriz $\nabla^2 f(x)$ ser simétrica, parece razoável exigir que cada aproximação à matriz Hessiana também o seja. Assim, a actualização deverá ser feita de forma que, se $B^{(k)}$ for simétrica então também $B^{(k+1)}$ o será. Deste modo, partindo de uma matriz simétrica garante-se a simetria de $B^{(k)}$ ao longo do processo iterativo. Para uma correcção de característica um, a manutenção da simetria define de modo único a matriz de correcção. Assim, para que $B^{(k+1)}$ seja simétrica (supondo que $B^{(k)}$ o é), v terá de ser um múltiplo de u , isto é,

$$v = \alpha u = \alpha(y^{(k)} - B^{(k)}s^{(k)}), \quad \alpha \neq 0.$$

A actualização de característica um (4.20) será então da forma

$$B^{(k+1)} = B^{(k)} + \frac{(y^{(k)} - B^{(k)}s^{(k)}) (y^{(k)} - B^{(k)}s^{(k)})^T}{(y^{(k)} - B^{(k)}s^{(k)})^T s^{(k)}}, \quad (4.21)$$

com $y^{(k)} - B^{(k)}s^{(k)}$ e $(y^{(k)} - B^{(k)}s^{(k)})^T s^{(k)}$ diferentes de zero, donde

$$C^{(k)} = \frac{(y^{(k)} - B^{(k)}s^{(k)}) (y^{(k)} - B^{(k)}s^{(k)})^T}{(y^{(k)} - B^{(k)}s^{(k)})^T s^{(k)}}$$

é a matriz de correcção. Esta actualização é denominada *actualização simétrica de característica um*.

Pelo facto da actualização simétrica de característica um ser única, para obter outras actualizações simétricas, será necessário permitir que a matriz de correcção tenha característica dois. Uma técnica para desenvolver actualizações simétricas consiste em, dada a matriz simétrica $B^{(k)}$, definir $B_0 = B^{(k)}$, e gerar uma actualização B_1 , usando a actualização de característica um (4.20), isto é,

$$B_1 = B_0 + uv^T, \quad v^T s^{(k)} \neq 0.$$

A matriz B_1 satisfaz a condição Quasi-Newton, não sendo no entanto simétrica, pelo que B_1 é transformada numa matriz simétrica B_2 , usando

$$B_2 = \frac{1}{2} (B_1 + B_1^T).$$

Contudo, visto que B_2 , em geral, não satisfaz a condição Quasi-Newton, repete-se o processo. Desta forma, gera-se uma sequência de matrizes de atualização

$$B_{2j+1} = B_{2j} + \frac{1}{v^T s^{(k)}} \left(y^{(k)} - B_{2j} s^{(k)} \right) v^T$$

e

$$B_{2j+2} = \frac{1}{2} \left(B_{2j+1} + B_{2j+1}^T \right)$$

com $j = 0, 1, 2, \dots$ [Gill–Murray–Wright, p.116].

A sequência tem um limite dado por

$$\begin{aligned} B^{(k+1)} = B^{(k)} + & \frac{\left(y^{(k)} - B^{(k)} s^{(k)} \right) v^T + v \left(y^{(k)} - B^{(k)} s^{(k)} \right)^T}{v^T s^{(k)}} - \\ & - \frac{\left(y^{(k)} - B^{(k)} s^{(k)} \right)^T s^{(k)}}{\left(v^T s^{(k)} \right)^2} v v^T. \end{aligned} \quad (4.22)$$

A atualização simétrica definida em (4.22) é de característica dois e fica bem definida para qualquer v que não seja ortogonal a $s^{(k)}$. A atualização simétrica de característica dois, análoga à de característica um, pode ser obtida fazendo $v = y^{(k)} - B^{(k)} s^{(k)}$.

Considerando $v = s^{(k)}$ obtém-se a chamada *atualização Powell-Symmetric-Broyden (PSB)*:

$$\begin{aligned} B^{(k+1)} = B^{(k)} + & \frac{\left(y^{(k)} - B^{(k)} s^{(k)} \right) s^{(k)T} + s^{(k)} \left(y^{(k)} - B^{(k)} s^{(k)} \right)^T}{s^{(k)T} s^{(k)}} - \\ & - \frac{\left(y^{(k)} - B^{(k)} s^{(k)} \right)^T s^{(k)}}{\left(s^{(k)T} s^{(k)} \right)^2} s^{(k)} s^{(k)T}. \end{aligned}$$

Quando $v = y^{(k)}$ obtém-se a conhecida *atualização Davidon-Fletcher-Powell (DFP)*:

$$\begin{aligned} B^{(k+1)} = B^{(k)} - & \frac{B^{(k)} s^{(k)} s^{(k)T} B^{(k)}}{s^{(k)T} B^{(k)} s^{(k)}} + \frac{y^{(k)} y^{(k)T}}{y^{(k)T} s^{(k)}} + \\ & + \left(s^{(k)T} B^{(k)} s^{(k)} \right) w^{(k)} w^{(k)T}, \end{aligned} \quad (4.23)$$

onde

$$w^{(k)} = \frac{y^{(k)}}{y^{(k)T} s^{(k)}} - \frac{B^{(k)} s^{(k)}}{s^{(k)T} B^{(k)} s^{(k)}}$$

e verifica-se, por substituição directa, que $w^{(k)}$ é ortogonal a $s^{(k)}$. Desta forma, qualquer múltiplo da matriz de característica um, $w^{(k)} w^{(k)T}$, pode ser adicionada a $B^{(k+1)}$, sendo que a matriz resultante continua a satisfazer a condição Quasi-Newton. Deste modo, obtemos uma família de actualizações

$$B_{\theta}^{(k+1)} = B^{(k)} - \frac{B^{(k)} s^{(k)} s^{(k)T} B^{(k)}}{s^{(k)T} B^{(k)} s^{(k)}} + \frac{y^{(k)} y^{(k)T}}{y^{(k)T} s^{(k)}} + \theta \left(s^{(k)T} B^{(k)} s^{(k)} \right) w^{(k)} w^{(k)T}, \quad (4.24)$$

com θ escalar. Considerando $\theta = 0$ obtém-se a denominada *actualização Broyden-Fletcher-Goldfarb-Shanno (BFGS)*, dada pela fórmula:

$$B^{(k+1)} = B^{(k)} - \frac{B^{(k)} s^{(k)} s^{(k)T} B^{(k)}}{s^{(k)T} B^{(k)} s^{(k)}} + \frac{y^{(k)} y^{(k)T}}{y^{(k)T} s^{(k)}}. \quad (4.25)$$

Se a direcção de procura for calculada a partir de

$$B^{(k)} d^{(k)} = -\nabla f(x^{(k)}),$$

é possível realizar uma importante simplificação a nível computacional na família de fórmulas de actualização (4.24), visto

$$B^{(k)} s^{(k)} = B^{(k)} (x^{(k+1)} - x^{(k)}) = \alpha^{(k)} B^{(k)} d^{(k)} = -\alpha^{(k)} \nabla f(x^{(k)}).$$

Considerando, por exemplo, a fórmula de actualização BFGS, dada em (4.25), obtemos

$$B^{(k+1)} = B^{(k)} + \frac{\nabla f(x^{(k)}) \nabla f(x^{(k)})^T}{\nabla f(x^{(k)})^T d^{(k)}} + \frac{y^{(k)} y^{(k)T}}{\alpha^{(k)} y^{(k)T} d^{(k)}}. \quad (4.26)$$

Positiva Definida

Uma vez que um ponto crítico de f é mínimo local forte se a matriz Hessiana for positiva definida nesse ponto, parece ser desejável que as matrizes de aproximação, $\{B^{(k)}\}$, sejam positivas definidas. Além disso, se $B^{(k)}$ é positiva definida, o modelo quadrático definido tem um mínimo local único, e a direcção de procura, $d^{(k)}$, calculada de $B^{(k)}d^{(k)} = -\nabla f(x^{(k)})$, é uma direcção descendente pois,

$$d^{(k)} = -\left(B^{(k)}\right)^{-1} \nabla f(x^{(k)})$$

donde

$$\nabla f(x^{(k)})^T d^{(k)} = -\nabla f(x^{(k)})^T \left(B^{(k)}\right)^{-1} \nabla f(x^{(k)}) < 0.$$

Assim, é usual requerer-se que a fórmula de actualização seja tal que se $B^{(k)}$ for positiva definida então também $B^{(k+1)}$ será positiva definida.

Teorema 4.4 *Se $B^{(k)}$ é positiva definida e $B^{(k+1)}$ é obtida usando a fórmula de actualização (4.25) (BFGS) então, se for implementada uma procura unidimensional exacta, $B^{(k+1)}$ também é positiva definida.*

Demonstração: A demonstração deste resultado pode ser vista em [Gill–Murray–Wright, p.120] \square

4.3.2 Fórmulas de Actualização Para a Inversa da Hessiana

Como já vimos, em certas situações, em vez de se ter uma aproximação à matriz Hessiana, seria vantajoso que se tivesse uma aproximação à matriz inversa da Hessiana. As correspondentes fórmulas de actualização deverão manter as propriedades citadas atrás. O resultado que se segue vai, conforme desejado, permitir determinar aproximações Quasi-Newton à inversa da matriz Hessiana, partindo das actualizações calculadas anteriormente.

Teorema 4.5 [Sherman-Morrison-Woodbury] *Sejam u e v vectores e A uma matriz não singular. Então $A + uv^T$ é não singular se e só se*

$$1 + v^T A^{-1} u \neq 0.$$

Nestas condições temos ainda

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} uv^T A^{-1}. \quad (4.27)$$

Demonstração: Ora, temos

$$\begin{aligned} & (A + uv^T) \left[A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} uv^T A^{-1} \right] = \\ = & I - \frac{1}{1 + v^T A^{-1} u} uv^T A^{-1} + uv^T A^{-1} - \frac{uv^T A^{-1} uv^T A^{-1}}{1 + v^T A^{-1} u} = \\ = & I - \left[\frac{1}{1 + v^T A^{-1} u} - 1 + \frac{uv^T A^{-1}}{1 + v^T A^{-1} u} \right] uv^T A^{-1} = \\ = & I. \end{aligned}$$

De modo análogo, mostra-se que

$$\left[A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} uv^T A^{-1} \right] (A + uv^T) = I$$

□

Exemplo 4.1: Consideremos a fórmula de actualização simétrica de característica um dada em (4.21). Simplificando a notação temos

$$\widehat{B} = B + \frac{(y - Bs)(y - Bs)^T}{(y - Bs)^T s}.$$

Pretendemos determinar \widehat{B}^{-1} . Ora, tendo em conta que B (e B^{-1}) é simétrica,

$$\begin{aligned} \widehat{B}^{-1} &= B^{-1} - \frac{B^{-1} \frac{(y - Bs)(y - Bs)^T}{(y - Bs)^T s} B^{-1}}{1 + \frac{(y - Bs)^T B^{-1} (y - Bs)}{(y - Bs)^T s}} \\ &= B^{-1} - \frac{B^{-1} (y - Bs)(y - Bs)^T B^{-1}}{(y - Bs)^T s + (y - Bs)^T B^{-1} (y - Bs)} \\ &= B^{-1} - \frac{(B^{-1} y - s)(y^T B^{-1} - s^T)}{y^T s - s^T B s + y^T B^{-1} y - y^T s - s^T y + s^T B s} \\ &= B^{-1} - \frac{(B^{-1} y - s)(B^{-1} y - s)^T}{y^T B^{-1} y - s^T y}. \end{aligned}$$

Obtemos desta forma a fórmula de actualização para aproximar inversas de Hessianas, pertence à classe das actualizações de característica um, devida a *Davidon*, e que tem por expressão

$$H^{(k+1)} = H^{(k)} - \frac{(H^{(k)}y - s)(H^{(k)}y - s)^T}{y^T H^{(k)}y - s^T y}$$

□

No exemplo seguinte, faz-se a dedução da fórmula BFGS para aproximar a inversa da matriz Hessiana, partindo da fórmula (4.25) para aproximação da matriz Hessiana.

Exemplo 4.2: Consideremos então a fórmula de actualização simétrica e definida positiva (se implementada uma pesquisa exacta) para a matriz Hessiana dada por (4.25). Por uma questão de simplificação da notação durante os cálculos, consideremos $B = B^{(k)}$ e $\hat{B} = B^{(k+1)}$ donde

$$\hat{B} = B - \frac{Bs(Bs)^T}{s^T Bs} + \frac{yy^T}{y^T s}$$

e pretendemos determinar \hat{B}^{-1} . Começemos por considerar

$$R = B + \frac{yy^T}{y^T s}$$

então

$$\hat{B} = R - \frac{Bs(Bs)^T}{s^T Bs}$$

e

$$R^{-1} = B^{-1} - \frac{B^{-1}y(B^{-1}y)^T}{y^T(s + B^{-1}y)}.$$

Donde

$$\hat{B}^{-1} = R^{-1} + \frac{1}{s^T Bs - (Bs)^T R^{-1} Bs} R^{-1} Bs(Bs)^T R^{-1}. \quad (4.28)$$

Agora,

$$\begin{aligned} s^T Bs - (Bs)^T R^{-1} Bs &= s^T Bs - s^T Bs + \frac{s^T yy^T s}{y^T(s + B^{-1}y)} = \\ &= \frac{s^T yy^T s}{y^T(s + B^{-1}y)}. \end{aligned} \quad (4.29)$$

Por outro lado,

$$R^{-1}Bs(Bs)^T R^{-1} = ss^T - \frac{ss^T yy^T B^{-1} + B^{-1}yy^T ss^T}{y^T(s + B^{-1}y)} + \frac{B^{-1}yy^T ss^T yy^T B^{-1}}{(y^T(s + B^{-1}y))^2} \quad (4.30)$$

Substituindo (4.29) e (4.30) em (4.28) obtém-se

$$\begin{aligned} \hat{B}^{-1} &= R^{-1} - B^{-1} + \\ &+ \frac{y^T(s + B^{-1}y)ss^T - ss^T yy^T B^{-1} - B^{-1}yy^T ss^T}{s^T yy^T s} + \\ &+ \frac{B^{-1}yy^T ss^T y^T B^{-1}}{s^T yy^T s y^T (s + B^{-1}y)}. \end{aligned} \quad (4.31)$$

Tendo em conta que $y^T s$, $s^T y$ e $y^T B^{-1}y$ são escalares e substituindo R^{-1} obtemos de (4.31)

$$\begin{aligned} \hat{B}^{-1} &= B^{-1} - \frac{B^{-1}y(B^{-1}y)^T}{y^T(s + B^{-1}y)} + \frac{ss^T}{y^T s} + \frac{s(y^T B^{-1}y)s^T}{s^T yy^T s} - \\ &- \frac{sy^T B^{-1}}{y^T s} - \frac{B^{-1}ys^T}{y^T s} + \frac{B^{-1}yy^T B^{-1}}{y^T(s + B^{-1}y)}. \end{aligned}$$

Considerando agora $H = B^{-1}$ temos

$$\begin{aligned} \hat{H} &= H - \frac{Hy(Hy)^T}{y^T(s + Hy)} + \frac{ss^T}{y^T s} + \frac{s(y^T Hy)s^T}{s^T yy^T s} - \\ &- \frac{sy^T H}{y^T s} - \frac{Hys^T}{y^T s} + \frac{Hy y^T H}{y^T(s + Hy)}. \end{aligned}$$

É fácil verificar que

$$\hat{H} = \left(I - \frac{sy^T}{s^T y} \right) H \left(I - \frac{ys^T}{y^T s} \right) + \frac{ss^T}{s^T y}$$

donde se conclui, introduzindo o índice da iteração, que

$$H^{(k+1)} = \left(I - \frac{s^{(k)} y^{(k)T}}{s^{(k)T} y^{(k)}} \right) H^{(k)} \left(I - \frac{y^{(k)} s^{(k)T}}{y^{(k)T} s^{(k)}} \right) + \frac{s^{(k)} s^{(k)T}}{s^{(k)T} y^{(k)}}. \quad (4.32)$$

□

A fórmula definida em (4.32) é uma fórmula de característica dois, para aproximar a inversa da matriz Hessiana, e é denominada por *BFGS*.

De modo análogo, podemos calcular a partir da fórmula (4.23) uma nova fórmula para aproximar a inversa da matriz Hessiana, que mantém a simetria e o facto de ser positiva definida, denominada tal como a inicial por *DFP* (*Davidon-Fletcher-Powell*), e dada pela expressão

$$H^{(k+1)} = H^{(k)} - \frac{H^{(k)}y^{(k)}y^{(k)T}H^{(k)}}{y^{(k)T}H^{(k)}y^{(k)}} + \frac{s^{(k)}s^{(k)T}}{s^{(k)T}y^{(k)}}.$$

4.4 Síntese

O problema de optimização não linear sem restrições nas variáveis é analisado, com mais detalhe, neste capítulo. Ficou patente a relação entre este problema e o tratado no capítulo anterior. No entanto, eles não podem ser considerados equivalentes. O cálculo, por um processo iterativo, dos zeros do vector gradiente da função objectivo, não garante a convergência para um mínimo. Outras condições têm de ser verificadas, nomeadamente assegurar que a matriz Hessiana, ou uma aproximação, seja definida positiva, e garantir um decréscimo do valor da função objectivo, ao longo do processo iterativo.

Os métodos analisados neste capítulo, começando pelo método de Newton, passando pelo método de segurança de Newton, com a recorrência à técnica de procura unidimensional e terminando com os Quasi-Newton, tentam responder, em parte, a algumas das questões levantadas relativas à convergência para a solução.

Capítulo 5

Optimização Com Restrições

Na sequência dos dois capítulos anteriores, vamos considerar agora o problema que consiste em minimizar uma função não linear sujeita a um conjunto de restrições, também não lineares, do tipo igualdade. Usando notação matemática, pode-se escrever

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.a.} \quad & c(x) = 0, \end{aligned} \tag{5.1}$$

onde $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e considera-se o caso em que $m \leq n$. Suponha-se que as seguintes hipóteses são verificadas:

- H1: f e c_i ($i = 1, \dots, m$) são continuamente diferenciáveis pelo menos até à segunda ordem;
- H2: A matriz Jacobiano de $c(x)$, $\nabla c(x)$, tem m linhas linearmente independentes.

5.1 Multiplicadores de Lagrange

Uma técnica usada para resolver problemas com restrições é conhecida como o *método dos multiplicadores de Lagrange*, sendo uma técnica de grande importância na teoria da programação não linear ([Beltrami],[Byrd] e [Glad]).

Considerando o problema (5.1), qualquer ponto solução x^* tem de satisfazer as restrições de (5.1), isto é, tem de ser um ponto admissível. Em

particular, se x^* for um ponto admissível e h um vector tal que $x^* + h$ também é admissível, então h diz-se uma *perturbação admissível*. Suponhamos que todas as perturbações admissíveis se podem escrever em função de uma variável $t \in \mathbb{R}$, tal que para $V > 0$,

$$h(t) = [h_1(t), h_2(t), \dots, h_n(t)]^T, \quad -V \leq t \leq V$$

com $h_i : \mathbb{R} \rightarrow \mathbb{R}$ funções continuamente diferenciáveis em $t = 0$,

$$h(0) = 0 \tag{5.2}$$

e

$$\left[\frac{dh(t)}{dt} \right]_{t=0} = \hat{h} \neq 0. \tag{5.3}$$

Então, visto $x^* + h(t)$ ser um ponto admissível, temos

$$c_i(x^* + h(t)) = 0, \quad -V \leq t \leq V \quad (i = 1, 2, \dots, m). \tag{5.4}$$

Calculando a derivada de (5.4) em ordem a t , no ponto $t = 0$, e tendo em conta (5.2) e (5.3), obtém-se

$$\nabla c_i(x^*)^T \hat{h} = 0 \quad (i = 1, 2, \dots, m). \tag{5.5}$$

Defina-se agora

$$\begin{aligned} \psi : [-V, V] &\rightarrow \mathbb{R} \\ t &\mapsto f(x^* + h(t)). \end{aligned}$$

Como $x^* + h(t)$ é admissível para $t \in [-V, V]$, ψ tem um mínimo sem restrições em $t = 0$ se x^* for uma solução de (5.1). Vem então que $\psi'(0) = 0$ e, tal como para ∇c , tendo em conta (5.2) e (5.3),

$$\nabla f(x^*)^T \hat{h} = 0. \tag{5.6}$$

Considere-se agora $\lambda = [\lambda_1, \dots, \lambda_m]^T$, um vector com m reais arbitrários. Então de (5.5) e (5.6) temos

$$\left(\nabla f(x^*) - \sum_{i=1}^m \lambda_i \nabla c_i(x^*) \right)^T \hat{h} = 0$$

ou

$$\sum_{j=1}^n \left\{ \frac{\partial f(x^*)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x^*)}{\partial x_j} \right\} h_j = 0 \quad (5.7)$$

Sabemos ainda que os h_j ($j = 1, \dots, n$) não são todos arbitrários, pois $x^* + h$ é um ponto admissível, ou seja,

$$c_i(x^* + h) = 0 \quad (i = 1, \dots, m). \quad (5.8)$$

Supondo que c_i ($i = 1, \dots, m$) são tais que as equações (5.8) permitem escrever m dos h_j como função dos restantes $n - m$, então $n - m$ dos h_j são independentes e (5.7) é verificada para valores arbitrários dos λ_i e para valores arbitrários de $n - m$ dos h_j . Como os λ_i são arbitrários, podemos escolhê-los por forma a que satisfaçam m relações da forma

$$\frac{\partial f(x^*)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x^*)}{\partial x_j} = 0. \quad (5.9)$$

Fixando estes valores dos λ_i ($i = 1, \dots, m$) temos de (5.7) e (5.9)

$$\sum_j \left\{ \frac{\partial f(x^*)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x^*)}{\partial x_j} \right\} h_j = 0, \quad (5.10)$$

onde o somatório em j é sobre os $n - m$ valores correspondentes aos $n - m$ valores independentes (donde arbitrários) dos h_j . Temos então de (5.10), para estes $n - m$ valores de j ,

$$\frac{\partial f(x^*)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x^*)}{\partial x_j} = 0. \quad (5.11)$$

De (5.9) e (5.11) obtém-se então

$$\frac{\partial f(x^*)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x^*)}{\partial x_j} = 0 \quad (j = 1, \dots, n). \quad (5.12)$$

Temos ainda a acrescentar a estas equações, as m equações

$$c_i(x^*) = 0 \quad (i = 1, \dots, m). \quad (5.13)$$

Assim, as equações (5.12) e (5.13) constituem um sistema de $n + m$ equações não lineares que podem ser, em princípio, resolvidas para as $n + m$

variáveis x_i^* ($i = 1, \dots, n$) e λ_j ($j = 1, \dots, m$) por forma a que se obtenha uma solução para o problema (5.1). Os λ_j são chamados *multiplicadores de Lagrange*.

Defina-se

$$L(x, \lambda) = f(x) - c(x)^T \lambda. \quad (5.14)$$

A função $L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ é chamada a *função Lagrangeana* correspondente ao problema não linear (5.1). Agora,

$$\frac{\partial L(x, \lambda)}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial c_i(x)}{\partial x_j} \quad (j = 1, \dots, n), \quad (5.15)$$

pelo que, usando (5.12),

$$\frac{\partial L(x^*, \lambda)}{\partial x_j} = 0 \quad (j = 1, \dots, n)$$

e

$$\frac{\partial L(x, \lambda)}{\partial \lambda_k} = c_k(x) \xrightarrow{(5.13)} \frac{\partial L(x^*, \lambda)}{\partial \lambda_k} = 0 \quad (k = 1, \dots, m).$$

Este processo sugere uma técnica para a resolução do problema (5.1). Assim, formada a função Lagrangeana e calculando as suas derivadas parciais em ordem aos x_i ($i = 1, 2, \dots, n$) e aos λ_j ($j = 1, 2, \dots, m$) basta resolver o sistema de equações não lineares resultante usando, por exemplo, um dos métodos descritos no Capítulo 3.

Se existe um par (x^*, λ^*) tal que

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

então chama-se a este par o *ponto Kuhn-Tucker (KT)* do problema (5.1). Se, além disso, para todo o u que verifique $\nabla c(x^*)u = 0$ vem

$$u^T \nabla_{xx}^2 L(x^*, \lambda^*) u > 0,$$

então o ponto x^* é solução do problema (5.1).

5.2 Funções de Penalização

Para resolver o problema não linear (5.1) pode-se construir uma função auxiliar e definir um problema de otimização sem restrições, cujo mínimo seja x^* , solução de (5.1), ou que esteja relacionado com este de um modo conhecido. Assim, o problema original pode ser transformado numa sequência de subproblemas sem restrições sendo Φ a função objectivo auxiliar do problema sem restrições.

Por outro lado, de um modo geral, o mínimo de f sem restrições será um ponto não admissível, ou f pode ser ilimitada inferiormente. Assim, a sequência de soluções dos subproblemas sem restrições converge para x^* somente se Φ inclui um termo que garante a admissibilidade. Uma possível escolha para esse termo é uma função que seja uma penalização positiva, tanto maior quanto maior for a violação da admissibilidade. Para ilustrar esta ideia temos, por exemplo, a função de penalização quadrática ou a função Lagrangeana aumentada descritas, respectivamente, nas Secções 5.2.1 e 5.2.2.

5.2.1 Função de Penalização Quadrática

A função de penalização quadrática é definida por

$$\Phi_Q(x, \rho) = f(x) + \frac{\rho}{2} c(x)^T c(x) = f(x) + \frac{\rho}{2} \|c(x)\|_2^2, \quad (5.16)$$

onde ρ é o chamado *parâmetro de penalização* e $\frac{\rho}{2} c(x)^T c(x)$ o *termo de penalização* ([Powell] e [Gill–Murray–Wright, p.208]). Caso existam restrições de desigualdade, $c(x)$ deverá ser substituída por $\hat{c}(x)$ que conterà as restrições que são violadas em x . Por convenção, todas as restrições de igualdade são consideradas como violadas.

Sob condições pouco restritivas, pode mostrar-se que existe $x^*(\rho)$, mínimo de (5.16), para o qual

$$\lim_{\rho \rightarrow \infty} x^*(\rho) = x^*.$$

Não se deve, no entanto, calcular x^* com a precisão necessária aumentando simplesmente ρ para um valor grande e realizando uma única minimização sem restrições de Φ_Q pois o número de condição da matriz Hessiana de Φ_Q em x^* aumenta à medida que ρ aumenta, com singularidade no limite.

Se o valor de ρ é muito grande até um algoritmo robusto terá dificuldades para calcular $x^*(\rho)$. Assim, usa-se uma sequência de problemas sem restrições

em que, o valor de ρ vai aumentando gradualmente e $x^*(\rho)$ é usado como ponto inicial para a minimização com o novo parâmetro de penalização, até que o critério de convergência, para o problema original, seja satisfeito.

Exemplo 5.1: Considere-se o problema

$$\begin{aligned} \min \quad & x^2 \\ \text{s.a.} \quad & x - 1 = 0. \end{aligned}$$

Para este problema, temos

$$\Phi_Q(x, \rho) = x^2 + \frac{\rho}{2}(x - 1)^2$$

e

$$x^*(\rho) = \frac{\rho}{\rho + 2}.$$

Claramente,

$$\lim_{\rho \rightarrow \infty} x^*(\rho) = x^* = 1.$$

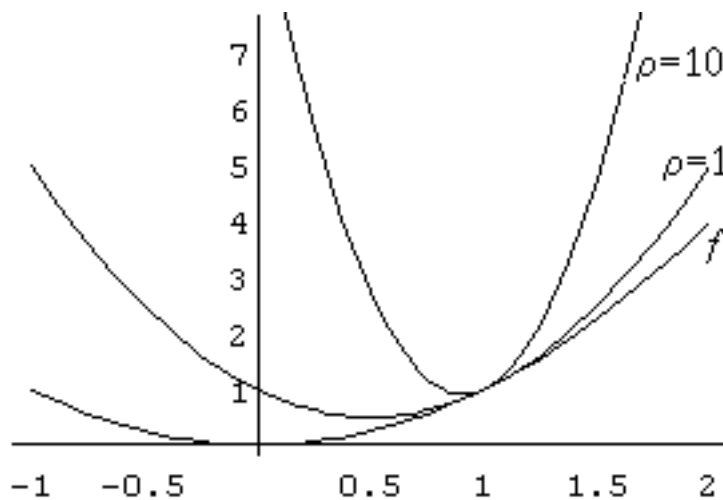


Figura 5.1: Gráfico de f e Φ_Q (com $\rho = 1$ e $\rho = 10$).

□

5.2.2 Função Lagrangeana Aumentada

Esta classe de funções pode ser obtida tendo em conta vários pontos de vista. Uma das motivações consiste em construir um subproblema sem restrições com uma função objectivo, Φ , para a qual o mau condicionamento seja evitável (em contraste com o descrito na Secção 5.2.1) mas mantendo a função a minimizar continuamente diferenciável.

A função Lagrangeana aumentada mais popular é

$$\Phi_A(x, \lambda, \rho) := f(x) - c(x)^T \lambda + \frac{\rho}{2} c(x)^T c(x), \quad (5.17)$$

onde, tal como na Secção 5.2.1, ρ é o parâmetro de penalização positivo. Uma propriedade importante desta função é o facto de que tanto em (5.17) como no seu gradiente, os termos de penalização anulam-se em x^* . Assim, se $\lambda = \lambda^*$, e tendo em conta (5.1), então x^* é um ponto crítico de (5.17) (em ordem a x).

Pode-se mostrar, sob condições pouco restritivas, que existe $\bar{\rho}$ tal que x^* é um mínimo sem restrições de $\Phi_A(x, \lambda^*, \rho)$, para todos $\rho > \bar{\rho}$. Desta forma, sob certas circunstâncias ideais, x^* poderia ser calculado realizando uma única minimização da função (5.17). Contudo, esta situação raramente ocorre, pois de um modo geral λ^* não está disponível até se ter encontrado a solução. Assim, os métodos que utilizam a Lagrangeana aumentada devem incluir um processo que estime o vector dos multiplicadores de Lagrange.

Façamos algumas observações:

- Obs. 1. Apenas podemos garantir que a solução x^* é um mínimo local de Φ_A . Além disso, Φ_A pode ser ilimitada inferiormente para alguns valores de ρ .
- Obs. 2. A escolha do parâmetro de penalização é, em geral, complicada mesmo quando Φ_A é ilimitada inferiormente. Obviamente que ρ deve ser suficientemente grande para que x^* seja um mínimo de Φ_A . Contudo, quando ele é muito grande ou muito pequeno podem surgir dificuldades. Quando ρ se torna demasiado grande ocorre o mau condicionamento dos subproblemas. Por outro lado, se o valor de ρ for muito pequeno existe não só a hipótese de Φ_A se tornar ilimitada inferiormente mas também a possibilidade de existir mau condicionamento do problema.

Obs. 3. O vector dos multiplicadores desempenha um papel fundamental pois, x^* não é necessariamente um ponto crítico de (5.17) excepto quando λ for igual a λ^* . Assim, o processo iterativo convergirá para x^* se as estimativas do vector dos multiplicadores também convergirem para λ^* . Pode-se mostrar que a razão de convergência das iterações $\{x_k\}$ não será melhor do que a razão de convergência das estimativas dos multiplicadores $\{\lambda_k\}$. Este resultado é bastante importante pois, mesmo que se aplique um método com convergência quadrática para resolver o problema $\min \Phi_A(x, \lambda, \rho)$, não se terá essa razão de convergência a menos que se use uma estimativa com a mesma razão de convergência para o vector dos multiplicadores.

Na próxima secção vamos introduzir uma técnica que é baseada na estimativa do vector dos multiplicadores e na sua substituição directa, na função que acabámos de referir.

5.3 Técnicas de Penalização Exacta

Certas funções de penalização definem *técnicas de penalização sequencial*, uma vez que o mínimo da função de penalização sem restrições converge para a solução do problema (5.1), x^* , apenas quando $\rho \rightarrow \infty$. Por outro lado, existem as *técnicas de penalização exacta* caracterizadas pelo facto de, existir um valor finito $\bar{\rho}$ tal que, para $\rho > \bar{\rho}$ o mínimo da função de penalização $\Phi(x, \lambda, \rho)$, $x^*(\rho)$, é igual a x^* . Como já foi referido na Secção 5.2.2, uma das funções para a qual se poderá implementar uma técnica de penalização exacta é a função Lagrangeana aumentada.

Assim, dada a função de penalização

$$\Phi_A(x, \lambda, \rho) = f(x) - c(x)^T \lambda + \frac{\rho}{2} c(x)^T c(x) \quad (5.18)$$

também já vimos que x^* é um seu mínimo se $\lambda = \lambda^*$. Isto sugere a ideia de introduzir a estimativa do vector dos multiplicadores na própria função Lagrangeana aumentada, como uma função contínua de x . Isto é, substitui-se λ por uma função $\lambda(x)$. À função $\lambda(x)$ chamamos *fórmula de substituição*. Um outro modo de estimar os mutiplicadores consiste em utilizar *fórmulas de actualização* ([Bertocci-Cavalli-Spedicato], [Gill-Murray-Wright], [Glad] e [Lucidi]). Estas fórmulas actualizam a estimativa, em cada iteração, por

exemplo, no final da resolução de cada subproblema (antes de incrementar o parâmetro de penalização). Um exemplo é a fórmula

$$\lambda^{(k+1)} = \lambda^{(k)} - \rho^{(k)} c(x^*(\rho^{(k)})),$$

em que $x^*(\rho^{(k)})$ foi o mínimo calculado do subproblema quando se utilizou $\lambda^{(k)}$ e $\rho^{(k)}$.

Todo o restante trabalho terá como base a função Lagrangeana aumentada e o uso de fórmulas de substituição.

Dada a função de penalização (5.18) construímos então uma nova função de penalização, que depende apenas do ponto x e do parâmetro de penalização ρ , obtendo-se

$$\Psi(x, \rho) = f(x) - c(x)^T \lambda(x) + \frac{\rho}{2} c(x)^T c(x). \quad (5.19)$$

As duas secções que se seguem são destinadas a deduzir e estudar as propriedades de duas dessas fórmulas de substituição.

5.3.1 Fórmula de Substituição $\lambda(x) = -\rho c(x)$

Considere-se a função Lagrangeana $L(x, \lambda) = f(x) - c(x)^T \lambda$, com $\lambda \in \mathbb{R}^m$ e $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Calculando o seu gradiente em ordem x , obtemos

$$\nabla_x L(x, \lambda) = \nabla f(x) - \nabla c(x)^T \lambda. \quad (5.20)$$

Considere-se, por outro lado, a função de penalização quadrática definida na Secção 5.2.1

$$\Phi_Q(x, \rho) = f(x) + \frac{\rho}{2} c(x)^T c(x).$$

Vem então

$$\nabla_x \Phi_Q(x, \rho) = \nabla f(x) + \rho \nabla c(x)^T c(x). \quad (5.21)$$

Como vimos, para que um ponto seja solução do problema (5.1), é necessário que ele seja ponto crítico da Lagrangeana em ordem a x . Pode-se provar que a solução de (5.1) é também ponto crítico de Φ_Q . Assim, as equações deverão obedecer a

$$\begin{cases} \nabla_x L(x, \lambda) = \nabla f(x) - \nabla c(x)^T \lambda & = 0 \\ \nabla \Phi_Q(x, \rho) = \nabla f(x) + \rho \nabla c(x)^T c(x) & = 0 \end{cases}$$

donde

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

e conseqüentemente

$$\nabla c(x)^T \lambda = -\rho \nabla c(x)^T c(x)$$

donde

$$\nabla c(x) \nabla c(x)^T \lambda = -\rho \nabla c(x) \nabla c(x)^T c(x).$$

De acordo com a hipótese H2, vem

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

donde

$$\lambda = -\rho c(x).$$

Podemos assim considerar a fórmula de substituição

$$\lambda(x) = -\rho c(x) \tag{5.22}$$

que tem como Jacobiano

$$\nabla \lambda(x) = -\rho \nabla c(x) \tag{5.23}$$

e Hessianas

$$\nabla^2 \lambda_i(x) = -\rho \nabla^2 c_i(x) \quad (i = 1, 2, \dots, m). \tag{5.24}$$

Para este caso, obtemos a Lagrangeana aumentada com o seguinte aspecto

$$\begin{aligned} \Psi_1(x, \rho) &= f(x) + \rho c(x)^T c(x) + \frac{\rho}{2} c(x)^T c(x) \\ &= f(x) + \frac{3}{2} \rho c(x)^T c(x) \end{aligned} \tag{5.25}$$

cujos gradiente e a Hessiana são respectivamente dados por

$$\nabla \Psi_1(x, \rho) = \nabla f(x) + 3\rho \nabla c(x)^T c(x) \tag{5.26}$$

e

$$\nabla^2 \Psi_1(x, \rho) = \nabla^2 f(x) + 3\rho \left(\nabla c(x)^T \nabla c(x) + \sum_{i=1}^m c_i(x) \nabla^2 c_i(x) \right). \tag{5.27}$$

5.3.2 Fórmula de Substituição $\lambda(x) = (\nabla c(x)^+)^T \nabla f(x)$

Vejam como deduzir outra fórmula de substituição, que recorre ao gradiente de f e ao Jacobiano de c .

Considere-se novamente a função Lagrangeana $L(x, \lambda) = f(x) - c(x)^T \lambda$ associada ao problema (5.1). Vem então que

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

e o ponto Kuhn-Tucker verifica

$$\nabla f(x) - \nabla c(x)^T \lambda = 0$$

donde

$$\nabla c(x)^T \lambda = \nabla f(x). \quad (5.28)$$

Considerando a hipótese H2 e a equação (5.28) verifica-se que

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

e consequentemente

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

Definindo $\nabla c(x)^+ = \nabla c(x)^T \left(\nabla c(x) \nabla c(x)^T\right)^{-1}$, como a inversa generalizada à direita da matriz $\nabla c(x)$, podemos então considerar a fórmula de substituição [Powell]

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x). \quad (5.29)$$

Esta fórmula de substituição tem uma propriedade muito importante que é indicada no seguinte teorema e que consiste no facto de que se (x^*, λ^*) for um ponto Kuhn-Tucker então $\lambda^* = \lambda(x^*)$.

Teorema 5.1 [Monteiro-Fernandes] *Suponhamos que se verificam as hipóteses H1 e H2 e que as condições de primeira ordem são satisfeitas no ponto (x^*, λ^*) , isto é, $\nabla_x L(x^*, \lambda^*) = 0$ e $\nabla_\lambda L(x^*, \lambda^*) = 0$. Então a função $\lambda(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ definida por*

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x)$$

satisfaz $\lambda^ = \lambda(x^*)$.*

Demonstração: Supondo então que (x^*, λ^*) satisfaz as condições de primeira ordem, vem $\nabla_x L(x^*, \lambda^*) = 0$ donde

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

A solução deste sistema de n equações com m incógnitas deve ser calculado pelo método dos mínimos quadrados. Assim,

$$\nabla c(x^*) \nabla c(x^*)^T \lambda^* = \nabla c(x^*) \nabla f(x^*),$$

ou seja, $\lambda^* = ((\nabla c(x^*))^+)^T \nabla f(x^*)$. Concluimos então, como era pretendido, que $\lambda^* = \lambda(x^*)$. \square

Substituindo $\lambda(x)$ de (5.29) em (5.19) obtém-se

$$\Psi_2(x, \rho) = f(x) - c(x)^T [(\nabla c(x)^+)^T \nabla f(x)] + \frac{\rho}{2} c(x)^T c(x),$$

ou seja, temos um problema de optimização sem restrições que depende apenas de x e ρ . O objectivo agora, é determinar a solução de

$$\min_{x \in \mathbb{R}^n} \Psi_2(x, \rho). \quad (5.30)$$

Para determinar a solução deste problema, pode-se por exemplo aplicar o método de Newton ou um Quasi-Newton. Na aplicação destes métodos, pode haver a necessidade de ajustar o parâmetro de penalização até que as condições de optimalidade se verifiquem.

Exemplo 5.2: Dado o problema de minimização com restrições

$$\begin{aligned} \min \quad & (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4 \\ \text{s.a.} \quad & x_1(1 + x_2^2) + x_3^4 - 8.2426 = 0 \end{aligned}$$

pode-se então escrever o problema equivalente sem restrições

$$\min_{(x, \lambda)} \Phi(x, \lambda, \rho) = f(x) - c(x)^T \lambda + \frac{\rho}{2} c(x)^T c(x)$$

com $(x, \lambda) \in \mathbb{R}^{3+1}$. Usando agora a fórmula de substituição (5.29), com

$$\nabla c(x) = \begin{bmatrix} 1 + x_2^2 & 2x_1x_2 & 4x_3^3 \end{bmatrix}$$

e

$$\nabla f(x) = \begin{bmatrix} 4x_1 - 2x_2 - 2 \\ -2(x_1 - x_2) + 4(x_2 - x_3)^3 \\ -4(x_2 - x_3)^3 \end{bmatrix},$$

vem

$$\begin{aligned} \lambda(x) &= (\nabla c(x)^+)^T \nabla f(x) = \\ &= \left((\nabla c(x) \nabla c(x)^T)^{-1} \nabla c(x) \nabla f(x) \right) = \\ &= \left((1 + x_2^2)^2 + (2x_1 x_2)^2 + (4x_3^3)^2 \right)^{-1} \times \\ &\quad \times [(4x_1 - 2x_2 - 2)(1 + x_2^2) + 2x_1 x_2 (-2(x_1 - x_2) + \\ &\quad + 4(x_2 - x_3)^3) - 16(x_2 - x_3)^3 x_3^3] \end{aligned}$$

Donde

$$\begin{aligned} \Psi_2(x, \rho) &= (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4 - \\ &\quad - \{ (4x_1 - 2x_2 - 2)(1 + x_2^2) + 2x_1 x_2 [-2(x_1 - x_2) + \\ &\quad + 4(x_2 - x_3)^3] - 16(x_2 - x_3)^3 x_3^3 \} \times \\ &\quad \times (2x_2^2 + 4x_1^2 x_2^2 + x_2^4 + 16x_3^6 + 1)^{-1} - \\ &\quad - \frac{\rho}{2} (x_1 + x_1 x_2^2 + x_3^4 - 8.2426)^2 \end{aligned}$$

□

Para aplicar o método de Newton a (5.30), surge a necessidade de determinar $\nabla \Psi_2(x, \rho)$ e $\nabla^2 \Psi_2(x, \rho)$ analiticamente (ou usando aproximações aos termos das derivadas). Analiticamente, obtém-se de (5.19)

$$\nabla \Psi_2(x, \rho) = \nabla f(x) - \nabla c(x)^T \lambda(x) - \nabla \lambda(x)^T c(x) + \rho \nabla c(x)^T c(x) \quad (5.31)$$

e

$$\begin{aligned} \nabla^2 \Psi_2(x, \rho) &= \nabla^2 f(x) - \sum_{i=1}^m \lambda_i(x) \nabla^2 c_i(x) - \nabla c(x)^T \nabla \lambda(x) - \\ &\quad - \sum_{i=1}^m c_i(x) \nabla^2 \lambda_i(x) - \nabla \lambda(x)^T \nabla c(x) + \\ &\quad + \rho \left[\nabla c(x)^T \nabla c(x) + \sum_{i=1}^m c_i(x) \nabla^2 c_i(x) \right], \end{aligned} \quad (5.32)$$

o que por sua vez implica a necessidade de calcular $\nabla \lambda(x)$ e $\nabla^2 \lambda(x)$.

Teorema 5.2 [Monteiro-Fernandes] *Suponhamos que se verificam as hipóteses H1 e H2, então $\lambda(x)$ definido por (5.29) é diferenciável, $\nabla\lambda(x)$ está bem definido e é dado por*

$$\begin{aligned} \nabla\lambda(x) &= (\nabla c(x)^+)^T \left[\nabla^2 f(x) - \sum_{i=1}^m \lambda_i(x) \nabla^2 c_i(x) \right] + \\ &+ (\nabla c(x) \nabla c(x)^T)^{-1} \left(\sum_{i=1}^m \nabla^2 c_i(x) v(x) e_i^T \right)^T \end{aligned} \quad (5.33)$$

em que $v(x) = \nabla f(x) - \nabla c(x)^T \lambda(x)$ e e_i é a coluna i da matriz identidade de ordem m .

Demonstração: De acordo com as hipóteses e com a equação (5.33), a matriz $\nabla c(x) \nabla c(x)^T$ é não singular, donde $\nabla c(x)^+$ existe e $\nabla\lambda(x)$ está bem definido.

Ora, considerar a equação

$$\nabla f(x) - \nabla c(x)^T \lambda(x) = 0$$

é equivalente a considerar

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

para algum $v(x) \in \mathbb{R}^n$, pois a característica de $\nabla c(x)$, tendo em conta a hipótese H2, é m , donde a solução do sistema é única e para $v(x) = 0$. Derivando ambas as equações em ordem a x obtém-se

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

Agora, multiplicando a primeira expressão à esquerda por $\nabla c(x)$ e substituindo na segunda chega-se a

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

Usando a segunda equação obtemos

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

e como $\nabla c(x)\nabla c(x)^T$ é não singular tem-se

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

que é equivalente a 5.33. □

5.3.3 Recurso a Pseudo-Derivadas

Como se pode verificar, o simples cálculo de $\nabla\lambda(x)$, necessário para determinar $\nabla\Psi_2$ e $\nabla^2\Psi_2$, exige o cálculo de $m+1$ ($\nabla^2 f(x)$ e $\nabla^2 c_i(x)$, $i = 1, 2, \dots, m$) Hessianas, ou seja, $(m+1)\frac{n(n+1)}{2}$ derivadas de segunda ordem. Considere-se a seguinte hipótese:

H3: O vector $c(x)$, das funções de restrição, é aproximado, numa vizinhança de \bar{x} , por um modelo linear

$$l(x) = l(\bar{x} + d) = c(\bar{x}) + \nabla c(\bar{x})d = c(\bar{x}) + \nabla c(\bar{x})(x - \bar{x}).$$

Usando a hipóteses H3 obtemos

$$\nabla l(x) = \nabla c(\bar{x})$$

e

$$\nabla^2 l(x) = 0,$$

originando uma aproximação a $\nabla\lambda(x)$ dada por

$$\mathcal{D}\lambda(x) = \left(\nabla c(x)^+\right)^T \nabla^2 f(x) = \left(\nabla c(x)\nabla c(x)^T\right)^{-1} \nabla c(x)\nabla^2 f(x), \quad (5.34)$$

à qual chamaremos *pseudo-Jacobiano de $\lambda(x)$* .

Como consequência,obtemos o vector das pseudo-derivadas de $\Psi_2(x, \rho)$, substituindo $\nabla\lambda(x)$ por $\mathcal{D}\lambda(x)$ no vector definido em (5.31), obtendo-se

$$\begin{aligned} \mathcal{D}\Psi_2(x, \rho) = & \nabla f(x) - \nabla c(x)^T \left(\nabla c(x)\nabla c(x)^T\right)^{-1} \nabla c(x)\nabla f(x) - \\ & - \left(\left(\nabla c(x)\nabla c(x)^T\right)^{-1} \nabla c(x)\nabla^2 f(x)\right)^T c(x) + \\ & + \rho\nabla c(x)^T c(x), \end{aligned} \quad (5.35)$$

conhecido por *pseudo-gradiente de Ψ_2* . Se x^* é um ponto KT, a matriz das segundas pseudo-derivadas é então definida por

$$\begin{aligned} \mathcal{D}^2\Psi_2(x, \rho) = & \nabla^2 f(x) - \\ & - \nabla c(x)^T \left(\left(\nabla c(x)\nabla c(x)^T\right)^{-1} \nabla c(x)\nabla^2 f(x)\right) - \\ & - \left(\left(\left(\nabla c(x)\nabla c(x)^T\right)^{-1} \nabla c(x)\nabla^2 f(x)\right)^T - \rho\nabla c(x)^T\right) \nabla c(x), \end{aligned} \quad (5.36)$$

que se obtém da expressão (5.32) substituindo $\nabla\lambda(x)$ por $\mathcal{D}\lambda(x)$, usando a hipótese H3 e desprezando o quarto termo da definição de $\nabla^2\Psi_2(x, \rho)$ (5.32) visto este tender para zero quando $x \rightarrow x^*$.

Teorema 5.3 [Monteiro-Fernandes] *Suponhamos que se verificam as hipóteses H1, H2 e H3. Se o par (x^*, λ^*) for um ponto KT do problema definido em (5.1) e se*

$$\lambda^* = \lambda(x^*) = (\nabla c(x^*)^+)^T \nabla f(x^*),$$

então x^ é ponto crítico de $\Psi_2(x, \rho)$ e raiz de $\mathcal{D}\Psi_2(x, \rho) = 0$. Por sua vez, qualquer raiz de $\mathcal{D}\Psi_2(x, \rho) = 0$ que verifique as restrições $(c(x) = 0)$ é ponto crítico de $\Psi_2(x, \rho)$.*

Demonstração: Da equação (5.31) tem-se, num ponto KT,

$$\begin{aligned}
\nabla\Psi_2(x^*, \rho) &= \nabla f(x^*) - \nabla c(x^*)^T \lambda(x^*) - \nabla \lambda(x^*)^T c(x^*) + \\
&\quad + \rho \nabla c(x^*)^T c(x^*) \\
&= \nabla f(x^*) - \nabla c(x^*)^T \lambda(x^*) \\
&= \nabla f(x^*) - \nabla c(x^*)^T \lambda^* = 0
\end{aligned}$$

de acordo com as condições de primeira ordem do problema definido em (5.1) e x^* é ponto crítico de $\Psi_2(x, \rho)$.

Como $\mathcal{D}\Psi_2(x, \rho)$ calculada em x^* origina

$$\begin{aligned}
\mathcal{D}\Psi_2(x^*, \rho) &= \nabla f(x^*) - \nabla c(x^*)^T \lambda(x^*) - \mathcal{D}\lambda(x^*)^T c(x^*) + \\
&\quad + \rho \nabla c(x^*)^T c(x^*) = \\
&= \nabla f(x^*) - \nabla c(x^*)^T \lambda(x^*) = 0,
\end{aligned}$$

então x^* é também raiz de $\mathcal{D}\Psi_2(x, \rho) = 0$.

Supondo agora que x^* é raiz de $\mathcal{D}\Psi_2(x, \rho) = 0$, então

$$\nabla f(x^*) - \nabla c(x^*)^T \lambda(x^*) - \mathcal{D}\lambda(x^*)^T c(x^*) + \rho \nabla c(x^*)^T c(x^*) = 0.$$

Como

$$\nabla\Psi_2(x, \rho) = \mathcal{D}\Psi_2(x, \rho) + \mathcal{D}\lambda(x)^T c(x) - \nabla\lambda(x)^T c(x)$$

tem-se que para qualquer raiz x^* de $\mathcal{D}\Psi_2(x, \rho) = 0$, que verifique $c(x^*) = 0$, $\nabla\Psi_2(x^*, \rho) = 0$. \square

5.4 Síntese

A partir dos dois capítulos anteriores foi fácil introduzir o problema de otimização não linear com restrições nas variáveis do tipo igualdade. Uma das técnicas, que pode ser implementada para a sua resolução, transforma o problema com restrições numa sequência de problemas sem restrições, e recorre posteriormente aos algoritmos para otimização sem restrições. Alguns destes algoritmos foram estudados no capítulo anterior. Este trabalho considera explicitamente este tipo de técnicas, conhecidas na literatura como técnicas de penalização. A mesma filosofia quando estendida a problemas com restrições de desigualdade origina as técnicas de pontos interiores, também conhecidas por técnicas de barreira.

A introdução de uma filosofia de substituição do vector dos multiplicadores de Lagrange, por fórmulas que são funções das variáveis do problema, traduz-se numa maior eficácia, uma vez que a técnica de penalização resulta exacta em vez de sequencial. No entanto, para algumas dessas fórmulas de substituição foi referida e ficou provada a complexidade computacional envolvida na nova formulação do problema. A necessidade de simplificar essas expressões originou várias aproximações. A implementação destas aproximações, para efeitos comparativos, bem como a descrição pormenorizada das fórmulas envolvidas em cada uma das versões, fazem parte do conteúdo do próximo capítulo.

Capítulo 6

Experiências Computacionais

6.1 Implementação

A implementação das cinco versões baseadas em fórmulas de substituição do vector dos multiplicadores de Lagrange, para a resolução do problema

$$\min \Psi(x, \rho) = f(x) - c(x)^T \lambda(x) + \frac{\rho}{2} c(x)^T c(x),$$

que passaremos a descrever, foi efectuada usando a linguagem de programação *C* (*Borland C++*), sobre o sistema operativo *Windows95*.

O computador utilizado foi um *PC 486-DX4 (100 Mhz)* com *16M RAM*.

Foram implementadas uma versão do método de Newton, uma versão Quasi-Newton, uma Pseudo-Newton e duas versões que combinam actualizações Quasi-Newton com o método de Newton, com o objectivo de resolver um conjunto variado de problemas. Estes problemas foram retirados das colecções de problemas de optimização não lineares com restrições de [Hock-Schittkowski] e [Celis-Dennis-Tapia]. Recorremos, de acordo com o estudo teórico feito no capítulo anterior, às duas fórmulas de substituição definidas em 5.3.1 e 5.3.2, para aproximar o vector dos multiplicadores de Lagrange na Lagrangeana aumentada. Passamos então a descrever as versões.

6.1.1 Versão Newton Simples

A fórmula de substituição usada foi

$$\lambda(x) = -\rho c(x)$$

e foi aplicado o Método de Segurança de Newton (Algoritmo 4.2) à resolução de

$$\min \Psi_1(x, \rho) = f(x) + \frac{3}{2}\rho c(x)^T c(x)$$

com

$$\nabla \Psi_1(x, \rho) = \nabla f(x) + 3\rho \nabla c(x)^T c(x)$$

e

$$\begin{aligned} \nabla^2 \Psi_1(x, \rho) = & \nabla^2 f(x) + \\ & + 3\rho \left(\nabla c(x)^T \nabla c(x) + \sum_{i=1}^m c_i(x) \nabla^2 c_i(x) \right) \end{aligned}$$

6.1.2 Versão Quasi-Newton(BFGS)

Foi usada a fórmula de substituição

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x)$$

e implementou-se o método Quasi-Newton (Secção 4.3) baseado na fórmula de actualização *BFGS* (4.32) para resolver

$$\min \Psi_2(x, \rho)$$

aproximando

$$\left(\nabla^2 \Psi_2(x^{(k)}, \rho) \right)^{-1}$$

pela matriz $H^{(k)}$ de tal forma que

$$d^{(k)} = -H^{(k)} \nabla \Psi_2(x^{(k)}, \rho).$$

Evita-se assim o cálculo de $\nabla^2 \lambda_i(x)$ presente na Hessiana da Lagrangeana aumentada, Ψ_2 . O cálculo das matrizes Hessianas $\nabla^2 c_i(x)$ ($i = 1, 2, \dots, m$) é feito para determinar $\nabla \lambda(x)$ (veja-se (5.33)) usado em $\nabla \Psi_2$ (5.31).

A actualização *BFGS* é

$$H^{(k+1)} = \left(I - \frac{s^{(k)} y^{(k)T}}{s^{(k)T} y^{(k)}} \right) H^{(k)} \left(I - \frac{y^{(k)} s^{(k)T}}{y^{(k)T} s^{(k)}} \right) +$$

$$+ \frac{s^{(k)} s^{(k)T}}{s^{(k)T} y^{(k)}}$$

em que

$$\begin{cases} y^{(k)} &= \nabla \Psi_2(x^{(k+1)}, \rho) - \nabla \Psi_2(x^{(k)}, \rho) \\ s^{(k)} &= x^{(k+1)} - x^{(k)} \end{cases} \quad (6.1)$$

e, consideramos como aproximação inicial à inversa da matriz Hessiana de Ψ_2 a matriz identidade, $H^{(0)} = I_n$.

6.1.3 Versão Pseudo-Newton

Nesta versão também se usa a fórmula de substituição

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x),$$

recorrendo-se ao método de Segurança de Newton (Algoritmo 4.2) e à utilização das pseudo-derivadas. Assim, $\nabla \Psi_2(x, \rho)$ e $\nabla^2 \Psi_2(x, \rho)$ são aproximadas, respectivamente, por $\mathcal{D}\Psi_2(x, \rho)$ e $\mathcal{D}^2\Psi_2(x, \rho)$ no sistema das equações Newton (veja-se (5.35) e (5.36)).

O sistema resultante, conhecido por pseudo-Newton, é agora

$$\mathcal{D}^2\Psi_2(x^{(k)}, \rho)d^{(k)} = -\mathcal{D}\Psi_2(x^{(k)}, \rho).$$

6.1.4 Versão Newton/Quasi-Newton(BFGS)

Continua a ser implementada a seguinte fórmula de substituição

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x).$$

No entanto, como $\nabla^2 \Psi_2$ depende de $\nabla^2 \lambda_i(x)$ (veja-se o quarto termo da definição de $\nabla^2 \Psi_2$, em (5.32)), esta versão aproxima as matrizes Hessianas $\nabla^2 \lambda_i(x)$ ($i = 1, 2, \dots, m$) por matrizes obtidas de através de actualizações do tipo Quasi-Newton. Seja $B_i^{(k)}$ a aproximação à matriz $\nabla^2 \lambda_i(x)$ obtida pela fórmula de actualização BFGS:

$$B_i^{(k+1)} = B_i^{(k)} - \frac{B_i^{(k)} s^{(k)} s^{(k)T} B_i^{(k)}}{s^{(k)T} B_i^{(k)} s^{(k)}} + \frac{y_i^{(k)} y_i^{(k)T}}{y_i^{(k)T} s^{(k)}}$$

em que

$$\begin{cases} y_i^{(k)} &= \nabla \lambda^{(i)}(x^{(k+1)}) - \nabla \lambda^{(i)}(x^{(k)}) \\ s^{(k)} &= x^{(k+1)} - x^{(k)} \end{cases}$$

onde $\nabla \lambda^{(i)}$ representa a linha i da matriz do Jacobiano $\nabla \lambda$, com

$$B_i^{(0)} = I, \quad i = 1, 2, \dots, m.$$

Temos então $\nabla^2 \Psi_2(x^{(k)}, \rho) \approx \bar{\mathcal{B}}(x^{(k)}, \rho)$, em que

$$\begin{aligned} \bar{\mathcal{B}}(x^{(k)}, \rho) &= \nabla^2 f(x^{(k)}) - \sum_{i=1}^m \lambda_i(x^{(k)}) \nabla^2 c_i(x^{(k)}) - \nabla c(x^{(k)})^T \nabla \lambda(x^{(k)}) - \\ &- \sum_{i=1}^m c_i(x^{(k)}) \boxed{\mathbf{B}_i^{(k)}} - [\nabla \lambda(x^{(k)}) - \rho \nabla c(x^{(k)})]^T \nabla c(x^{(k)}) + \\ &+ \rho \sum_{i=1}^m c_i(x^{(k)}) \nabla^2 c_i(x^{(k)}). \end{aligned}$$

O sistema Newton fica com a forma

$$\bar{\mathcal{B}}(x^{(k)}, \rho) d^{(k)} = -\nabla \Psi_2(x^{(k)}, \rho),$$

sendo usado, na sua resolução, o *método de Segurança de Newton* (Algoritmo 4.2).

6.1.5 Versão Newton/Quasi-Newton(BFGS/Broyden)

A fórmula de substituição utilizada foi:

$$\lambda(x) = (\nabla c(x)^+)^T \nabla f(x).$$

Assim, lembrando,

$$\nabla \Psi_2(x, \rho) = \nabla f(x) - \nabla c(x)^T \lambda(x) - \boxed{\nabla \lambda(x)}^T c(x) + \rho \nabla c(x)^T c(x)$$

e

$$\begin{aligned} \nabla^2 \Psi_2(x, \rho) &= \nabla^2 f(x) - \sum_{i=1}^m \lambda_i(x) \nabla^2 c_i(x) - \nabla c(x)^T \boxed{\nabla \lambda(x)} - \\ &- \sum_{i=1}^m c_i(x) \boxed{\nabla^2 \lambda_i(x)} - \left[\boxed{\nabla \lambda(x)} - \rho \nabla c(x) \right]^T \nabla c(x) + \\ &+ \rho \sum_{i=1}^m c_i(x) \nabla^2 c_i(x), \end{aligned}$$

implementámos o algoritmo de Segurança de Newton (Algoritmo 4.2) ao sistema Newton. Nesta versão o sistema Newton é aproximado por

$$\widehat{\mathcal{B}}(x^{(k)}, \rho)d^{(k)} = -\widehat{a}(x^{(k)}, \rho),$$

em que $\widehat{\mathcal{B}}$ é uma aproximação à matriz Hessiana de Ψ_2 e \widehat{a} uma aproximação ao gradiente de Ψ_2 , definidas respectivamente por

$$\begin{aligned} \nabla\Psi_2(x^{(k)}, \rho) \approx \widehat{a}(x^{(k)}, \rho) &= \nabla f(x^{(k)}) - \nabla c(x^{(k)})^T \lambda(x^{(k)}) - \\ &- \boxed{\mathbf{A}(\mathbf{k})} c(x^{(k)}) + \rho \nabla c(x^{(k)})^T c(x^{(k)}) \end{aligned}$$

e

$$\begin{aligned} \nabla^2\Psi_2(x^{(k)}, \rho) \approx \widehat{\mathcal{B}}(x^{(k)}, \rho) &= \nabla^2 f(x^{(k)}) - \sum_{i=1}^m \lambda_i(x^{(k)}) \nabla^2 c_i(x^{(k)}) - \\ &- \nabla c(x^{(k)})^T \boxed{\mathbf{A}(\mathbf{k})} - \sum_{i=1}^m c_i(x^{(k)}) \boxed{\mathbf{B}_i(\mathbf{k})} - \\ &- \left[\boxed{\mathbf{A}(\mathbf{k})} - \rho \nabla c(x^{(k)}) \right]^T \nabla c(x^{(k)}) + \\ &+ \rho \sum_{i=1}^m c_i(x^{(k)}) \nabla^2 c_i(x^{(k)}). \end{aligned}$$

A fórmula de actualização usada para aproximar o Jacobiano de $\lambda(x)$ é a de Broyden (veja-se (3.13))

$$A^{(k+1)} = A^{(k)} + \frac{(y^{(k)} - A^{(k)}s^{(k)})s^{(k)T}}{s^{(k)T}s^{(k)}}$$

em que

$$\begin{cases} y^{(k)} &= \lambda(x^{(k+1)}) - \lambda(x^{(k)}) \\ s^{(k)} &= x^{(k+1)} - x^{(k)} \end{cases} \quad (6.2)$$

com $A^{(0)} = [I_m|0]$. Para aproximar as Hessianas dos λ_i ($i = 1, 2, \dots, m$) usou-se a mesma fórmula de actualização de BFGS, que foi usada na versão anterior.

6.2 Critério de Paragem

Quando se está a implementar um processo iterativo, com o intuito de obter uma sequência de aproximações à solução de um problema, deverá ter-se um critério que pare o processo com garantias de se estar próximo da solução.

Com precisão infinita, a condição necessária para que x seja um minimizante de Ψ é $\nabla\Psi(x) = 0$. No entanto, num processo iterativo com precisão finita necessitamos de alterar a condição para $\nabla\Psi(x) \approx 0$. Apesar de $\nabla\Psi(x) = 0$ também ocorrer para um ponto sela ou para um máximo, as estratégias de globalização e de perturbação da matriz Hessiana da função objectivo de modo a que seja definida positiva, torna a convergência para um máximo ou para um ponto sela virtualmente impossível. Assim, no nosso contexto, $\nabla\Psi(x, \rho) = 0$ é considerada uma condição necessária e suficiente para que x seja um minimizante local de $\Psi(x, \rho)$, ou seja, $\|\nabla\Psi(x, \rho)\| \leq \epsilon_1$ com ϵ_1 uma quantidade positiva e pequena.

Outras condições deverão ter em conta, nomeadamente, o facto da solução ter de pertencer à região admissível, o que utilizando precisão finita será traduzido por $\|c(x)\| \leq \epsilon_2$.

Foi então considerado o seguinte algoritmo para testar a convergência do processo iterativo:

Algoritmo 6.1: [Critério de Paragem]

Dados $\epsilon_1, \epsilon_2, \epsilon_3$ e ϵ_4 quantidades positivas e pequenas.

- Fazer *convergiu:=Falso*
- Se $\|c(x)\|_2 \leq \epsilon_1$ então
 - Se $\|x^{(k+1)} - x^{(k)}\|_2 \leq \epsilon_2$ e $\|\Psi(x^{(k+1)}) - \Psi(x^{(k)})\|_2 \leq \epsilon_3$ então
 - * Se $\|\nabla\Psi(x^{(k+1)})\|_2 \leq \epsilon_4$ então
 - Fazer *convergiu:=Verdade*

6.3 Parâmetro de Penalização

Como já foi referido, e se pode concluir dos resultados computacionais, o parâmetro de penalização tem grande influência nos resultados obtidos. Não sendo consensual a melhor solução para o obter, optou-se por um esquema em que o delimitamos inferiormente por ρ_{\min} , quando se está numa vizinhança δ

da região admissível, e superiormente por ρ_{\max} , quando se está longe da região admissível, traduzido por $\|c(x)\|_2 \geq \Delta$. Noutras situações, ρ é calculado por uma interpolação linear baseada em (δ, ρ_{\min}) e (Δ, ρ_{\max}) .

Algoritmo 6.2: [Parâmetro de Penalização]

Dados $\delta, \Delta, \rho_{\max}$ e ρ_{\min}

- Se $\|c(x)\|_2 \leq \delta$ então

- Fazer $\rho := \rho_{\min}$

senão

- Se $\|c(x)\|_2 \geq \Delta$ então

- * Fazer $\rho := \rho_{\max}$

senão

- * Fazer $\rho := \frac{\rho_{\max}(\|c(x)\|_2 - \delta) + \rho_{\min}(\Delta - \|c(x)\|_2)}{\Delta - \delta}$

6.4 Resultados Computacionais

Para simplificar a interpretação dos resultados obtidos, denotaremos a versão Newton Simples, em (6.1.1), por versão 1, a versão Quasi-Newton (BFGS), em (6.1.2), por versão 2 a versão Pseudo-Newton, em (6.1.3), por versão 3 e as versões Newton/Quasi-Newton, em (6.1.4) e (6.1.5) por versão 4 e versão 5, respectivamente.

Foram testados trinta e um problemas das colecções atrás referidas, consideraram-se os seguintes valores para as constantes dos Algoritmos 6.1 e 6.2: $\epsilon_1 = 10^{-2}$, $\epsilon_2 = \epsilon_3 = \epsilon_4 = 10^{-6}$, $\delta = 10^{-2}$, $\Delta \in \{1, 10\}$, $\rho_{\min} = 1.0$ e $\rho_{\max} \in \{5, 50, 500\}$, obtendo-se um total de 186 tentativas (6 para cada um dos problema). Considerou-se ainda que o número máximo de iterações permitido fosse 1000. Falta referir que a técnica de procura unidimensional implementada foi o critério de *Goldstein-Armijo* (veja-se na Secção 4.2 as condições (4.12) e (4.13)) considerando $\mu_1 = 0.9$ e $\mu_2 = 10^{-3}$.

Do conjunto de problemas testados, os problemas 7, 39, 55 e 62 da colecção [Hock-Schittkowski] não convergiram, em nenhuma das cinco versões, por terem atingido o número máximo de iterações permitido (7, 39, 55 e 62) ou por ter ocorrido *overflow* (55 e 62) ou pelo facto de, no processo iterativo, se ter obtido um ponto que não pertencia ao domínio de uma das funções envolvidas (62). A partir de agora vamos apenas considerar os restantes vinte e sete problemas (num total de 162 tentativas).

Assim, da compilação de exercícios referidos, a versão 1 foi a que resolveu com sucesso o menor número de problemas. Por outro lado, as versões 3, 4 e 5 resolveram, respectivamente, 71,0%, 73,5% e 72,8% das 162 tentativas. Note-se, no entanto, que a versão 5 conseguiu convergir nos vinte e sete problemas que estamos a considerar (para alguma combinação de (ρ_{\max}, Δ)).

As percentagens de sucessos das cinco versões encontram-se na Tabela 6.1.

	Versão 1	Versão 2	Versão 3	Versão 4	Versão 5
Sucessos nos 27 problemas ¹ (%)	10 (37,0%)	24 (88,9%)	20 (74,1%)	24 (88,9%)	27 (100,0%)
Sucessos nas 162 tentativas (%)	55 (34,0%)	108 (66,7%)	115 (71,0%)	119 (73,5%)	118 (72,8%)

Tabela 6.1: Número total (e percentagem) de sucessos.

Nas tabelas seguintes são apresentados os resultados obtidos com maior pormenor. A notação das referidas tabelas é a seguinte:

- V – Versão.
- Nit – Número de iterações até convergir, de acordo com o critério de paragem referido na secção 6.2.
- Tmp – Tempo de *CPU*, em milésimos de segundo, que demorou a realizar as Nit iterações.
- N_{\square} – Número de cálculos da função \square .
- ρ_{\max} e Δ – Constantes usadas para determinar o valor do parâmetro de penalização, de acordo com o referido no Algoritmo 6.2.
- Ov – Ocorreu um *overflow* durante o processo iterativo.
- $> 10^3$ – O processo iterativo não convergiu em 1000 iterações.

Na legenda das tabelas é indicado o número do problema, a colecção a que pertence (†-[Hock-Schittkowski] e ††-[Celis-Dennis-Tapia]), a categoria e as suas dimensões (n -número de variáveis e m -número de restrições).

¹Pelo menos um sucesso entre as seis combinações do par (ρ_{\max}, Δ) .

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	2	0e+00	0e+00	388	5	3	5	2	5	1
2	9	1e-07	5e-08	433	20	20	19	18	5	1
3	2	0e+00	0e+00	390	5	5	5	0	5	1
4	2	0e+00	0e+00	388	5	5	5	4	5	1
5	12	2e-08	3e-07	490	47	47	25	35	5	1
1	2	0e+00	0e+00	390	5	3	5	2	5	10
2	9	1e-07	5e-08	441	20	20	19	18	5	10
3	2	0e+00	0e+00	390	5	5	5	0	5	10
4	2	0e+00	0e+00	380	5	5	5	4	5	10
5	14	7e-09	4e-09	488	33	33	29	41	5	10
1	2	0e+00	0e+00	390	5	3	5	2	50	1
2	9	1e-07	5e-08	380	20	20	19	18	50	1
3	2	0e+00	0e+00	380	5	5	5	0	50	1
4	2	0e+00	0e+00	390	5	5	5	4	50	1
5	11	9e-07	1e-07	490	23	23	23	32	50	1
1	2	0e+00	0e+00	380	5	3	5	2	50	10
2	9	1e-07	5e-08	441	20	20	19	18	50	10
3	2	0e+00	0e+00	380	5	5	5	0	50	10
4	2	0e+00	0e+00	439	5	5	5	4	50	10
5	12	2e-07	2e-08	490	47	47	25	35	50	10
1	2	0e+00	0e+00	378	5	3	5	2	500	1
2	9	1e-07	5e-08	439	20	20	19	18	500	1
3	2	0e+00	0e+00	390	5	5	5	0	500	1
4	2	0e+00	0e+00	390	5	5	5	4	500	1
5	12	2e-08	1e-08	441	25	25	25	35	500	1
1	2	0e+00	0e+00	380	5	3	5	2	500	10
2	9	1e-07	5e-08	439	20	20	19	18	500	10
3	2	0e+00	0e+00	380	5	5	5	0	500	10
4	2	0e+00	0e+00	390	5	5	5	4	500	10
5	11	9e-07	1e-07	441	23	23	23	32	500	10

Tabela 6.2: Resultados do problema 28[†] [QLRT](n=3,m=1)

As Versões 1, 3 e 4 foram as que apresentaram melhores resultados, sendo estes semelhantes. A versão 5 foi a que efectuou maior esforço computacional, seguida da versão Quasi-Newton.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	2	2e-30	0e+00	378	5	3	5	4	5	1
2	19	4e-08	1e-08	550	54	54	39	76	5	1
3	2	4e-30	0e+00	380	5	5	5	0	5	1
4	7	2e-16	4e-17	500	24	24	15	28	5	1
5	10	2e-09	2e-09	539	47	47	21	58	5	1
1	2	2e-30	0e+00	390	5	3	5	4	5	10
2	19	4e-07	6e-08	550	67	67	39	76	5	10
3	2	4e-30	0e+00	378	5	5	5	0	5	10
4	9	2e-32	0e+00	500	29	29	19	36	5	10
5	11	6e-08	4e-08	548	37	37	23	64	5	10
1	2	2e-30	0e+00	388	5	3	5	4	50	1
2	24	1e-07	3e-08	599	76	76	49	96	50	1
3	2	4e-30	0e+00	378	5	5	5	0	50	1
4	5	1e-23	0e+00	500	13	13	11	20	50	1
5	10	4e-08	1e-08	490	34	34	21	58	50	1
1	2	2e-30	0e+00	378	5	3	5	4	50	10
2	20	1e-07	4e-08	609	57	57	41	80	50	10
3	2	4e-30	0e+00	378	5	5	5	0	50	10
4	8	3e-23	0e+00	490	25	25	17	32	50	10
5	9	3e-08	2e-08	500	27	27	19	52	50	10
1	2	2e-30	0e+00	378	5	3	5	4	500	1
2	28	3e-07	1e-07	660	83	83	57	112	500	1
3	2	4e-30	0e+00	380	5	5	5	0	500	1
4	5	2e-27	0e+00	439	13	13	11	20	500	1
5	10	2e-08	6e-09	550	30	30	21	58	500	1
1	2	2e-30	0e+00	378	5	3	5	4	500	10
2	24	5e-07	1e-07	599	74	74	49	96	500	10
3	2	4e-30	0e+00	390	5	5	5	0	500	10
4	5	2e-25	0e+00	439	13	13	11	20	500	10
5	10	5e-08	2e-08	490	33	33	21	58	500	10

Tabela 6.3: Resultados do problema 48[†] [QLRT](n=5,m=2)

As versões 1 e 3 foram as que apresentaram melhores resultados. A versão 2 foi a que efectuou maior esforço computacional.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	2	0e+00	0e+00	388	5	3	5	6	5	1
2	6	2e-10	9e-11	439	15	15	13	36	5	1
3	2	0e+00	0e+00	441	5	5	5	0	5	1
4	2	0e+00	0e+00	390	5	5	5	12	5	1
5	$> 10^3$	2e+00	1e+00						5	1
1	2	0e+00	0e+00	388	5	3	5	6	5	10
2	6	2e-10	9e-11	439	15	15	13	36	5	10
3	2	0e+00	0e+00	388	5	5	5	0	5	10
4	2	0e+00	0e+00	378	5	5	5	12	5	10
5	$> 10^3$	5e+00	2e+00						5	10
1	2	0e+00	0e+00	380	5	3	5	6	50	1
2	6	2e-10	9e-11	439	15	15	13	36	50	1
3	2	0e+00	0e+00	390	5	5	5	0	50	1
4	2	0e+00	0e+00	390	5	5	5	12	50	1
5	$> 10^3$	2e-02	8e-03						50	1
1	2	0e+00	0e+00	388	5	3	5	6	50	10
2	6	2e-10	9e-11	439	15	15	13	36	50	10
3	2	0e+00	0e+00	388	5	5	5	0	50	10
4	2	0e+00	0e+00	429	5	5	5	12	50	10
5	$> 10^3$	1e+00	2e-01						50	10
1	2	0e+00	0e+00	388	5	3	5	6	500	1
2	6	2e-10	9e-11	439	15	15	13	36	500	1
3	2	0e+00	0e+00	378	5	5	5	0	500	1
4	2	0e+00	0e+00	439	5	5	5	12	500	1
5	130	1e-07	6e-06	3949	1158	1158	261	1164	500	1
1	2	0e+00	0e+00	378	5	3	5	6	500	10
2	6	2e-10	9e-11	439	15	15	13	36	500	10
3	2	0e+00	0e+00	388	5	5	5	0	500	10
4	2	0e+00	0e+00	378	5	5	5	12	500	10
5	$> 10^3$	7e-03	3e-03						500	10

Tabela 6.4: Resultados do problema 51[†] [QLRT](n=5,m=3)

Foi com as versões 1, 3 e 4 que se obtiveram os melhores resultados. A versão 5 convergiu uma única vez. A versão 2 foi também constante para as combinações dos parâmetros ρ_{\max} e Δ consideradas.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	3e-16	5e-01						5	1
2	456	7e-08	3e-08	11970	3950	3950	913	2736	5	1
3	2	2e-13	1e-14	441	5	5	5	0	5	1
4	$> 10^3$	4e+05	4e+04						5	1
5	$> 10^3$	8e+05	3e+04						5	1
1	$> 10^3$	7e-16	8e-01						5	10
2	67	4e-09	4e-10	2589	925	925	135	402	5	10
3	2	2e-13	1e-14	441	5	5	5	0	5	10
4	$> 10^3$	4e+05	4e+04						5	10
5	$> 10^3$	3e+05	1e+04						5	10
1	$> 10^3$	3e-15	2e-01						50	1
2	154	4e-07	2e-07	4330	1417	1417	309	924	50	1
3	2	2e-13	1e-14	390	5	5	5	0	50	1
4	10	3e-15	5e-16	660	91	91	21	60	50	1
5	$> 10^3$	7e-02	3e-02						50	1
1	$> 10^3$	2e-15	5e-01						50	10
2	187	4e-07	4e-08	3898	950	950	375	1122	50	10
3	2	2e-13	1e-14	378	5	5	5	0	50	10
4	$> 10^3$	5e+03	3e+02						50	10
5	$> 10^3$	1e+01	1e+00						50	10
1	$> 10^3$	5e-15	7e-02						500	1
2	100	8e-08	4e-08	2701	747	747	201	600	500	1
3	2	2e-13	1e-14	390	5	5	5	0	500	1
4	5	1e-15	4e-16	441	11	11	11	30	500	1
5	273	8e-07	8e-05	8019	2463	2463	547	2448	500	1
1	$> 10^3$	2e-15	2e-01						500	10
2	545	1e-08	2e-09	12580	3809	3809	1091	3270	500	10
3	2	2e-13	1e-14	380	5	5	5	0	500	10
4	5	2e-15	3e-16	439	11	11	11	30	500	10
5	$> 10^3$	2e-01	7e-02						500	10

Tabela 6.5: Resultados do problema 52[†] [QLRT](n=5,m=3)

A versão 1 não convergiu em nenhuma tentativa. Por outro lado, a versão 2 convergiu em todas as tentativas. A versão 3 foi a melhor para este problema, seguida da versão 4, e a versão 5 convergiu uma única vez.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	36	2e-15	2e-16	548	73	37	73	72	5	1
2	126	5e-09	1e-09	2138	599	599	253	504	5	1
3	36	8e-16	2e-16	660	73	73	73	0	5	1
4	36	8e-16	2e-16	818	73	73	73	144	5	1
5	41	3e-15	6e-16	878	109	109	83	244	5	1
1	36	2e-15	2e-16	490	73	37	73	72	5	10
2	126	3e-08	5e-09	2140	600	600	253	504	5	10
3	36	8e-16	2e-16	718	73	73	73	0	5	10
4	36	8e-16	2e-16	830	73	73	73	144	5	10
5	51	7e-09	2e-09	1150	249	249	103	304	5	10
1	36	2e-15	2e-16	548	73	37	73	72	50	1
2	126	7e-07	1e-07	2140	617	617	253	504	50	1
3	36	8e-16	2e-16	660	73	73	73	0	50	1
4	36	8e-16	2e-16	818	73	73	73	144	50	1
5	15	2e-08	4e-09	550	36	36	31	88	50	1
1	36	2e-15	2e-16	550	73	37	73	72	50	10
2	126	8e-09	2e-09	2080	600	600	253	504	50	10
3	36	8e-16	2e-16	660	73	73	73	0	50	10
4	36	8e-16	2e-16	820	73	73	73	144	50	10
5	42	5e-15	1e-15	878	116	116	85	250	50	10
1	36	2e-15	2e-16	548	73	37	73	72	500	1
2	126	4e-07	7e-08	2080	595	595	253	504	500	1
3	36	8e-16	2e-16	660	73	73	73	0	500	1
4	36	8e-16	2e-16	830	73	73	73	144	500	1
5	28	4e-09	9e-10	720	61	61	57	166	500	1
1	36	2e-15	2e-16	548	73	37	73	72	500	10
2	126	7e-07	1e-07	2089	609	609	253	504	500	10
3	36	8e-16	2e-16	660	73	73	73	0	500	10
4	36	8e-16	2e-16	820	73	73	73	144	500	10
5	$> 10^3$	5e+22	2e+10						500	10

Tabela 6.6: Resultados do problema 49[†] [PLRT](n=5,m=2)

As versões 1, 2, 3 e 4 efectuaram o mesmo esforço computacional em todas as tentativas. A versão 5, variou em todas as tentativas, conseguindo para $\rho_{\max} = 50$ e $\Delta = 1$ obter o menor número de iterações. Note-se que a versão 1, ao efectuar mais do dobro das iterações que a versão 5 efectuou na melhor tentativa, obteve tempos semelhantes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	10	0e+00	0e+00	439	21	11	21	30	5	1
2	15	2e-08	4e-09	548	46	46	31	90	5	1
3	9	0e+00	0e+00	490	19	19	19	0	5	1
4	9	0e+00	0e+00	500	19	19	19	54	5	1
5	$> 10^3$	8e+51	4e+50						5	1
1	10	0e+00	0e+00	390	21	11	21	30	5	10
2	15	2e-08	4e-09	550	46	46	31	90	5	10
3	9	0e+00	0e+00	490	19	19	19	0	5	10
4	9	0e+00	0e+00	500	19	19	19	54	5	10
5	$> 10^3$	8e+51	4e+50						5	10
1	10	0e+00	0e+00	439	21	11	21	30	50	1
2	15	2e-08	4e-09	601	46	46	31	90	50	1
3	9	0e+00	0e+00	439	19	19	19	0	50	1
4	9	0e+00	0e+00	550	19	19	19	54	50	1
5	20	8e-08	1e-08	660	43	43	41	177	50	1
1	10	0e+00	0e+00	429	21	11	21	30	50	10
2	15	2e-08	4e-09	550	46	46	31	90	50	10
3	9	0e+00	0e+00	500	19	19	19	0	50	10
4	9	0e+00	0e+00	500	19	19	19	54	50	10
5	17	3e-07	6e-08	658	40	40	35	150	50	10
1	10	0e+00	0e+00	380	21	11	21	30	500	1
2	15	2e-08	4e-09	599	46	46	31	90	500	1
3	9	0e+00	0e+00	439	19	19	19	0	500	1
4	9	0e+00	0e+00	548	19	19	19	54	500	1
5	17	3e-07	2e-08	599	37	37	35	150	500	1
1	10	0e+00	0e+00	439	21	11	21	30	500	10
2	15	2e-08	4e-09	601	46	46	31	90	500	10
3	9	0e+00	0e+00	441	19	19	19	0	500	10
4	9	0e+00	0e+00	550	19	19	19	54	500	10
5	18	9e-07	2e-07	599	45	45	37	159	500	10

Tabela 6.7: Resultados do problema 50[†] [PLRT](n=5,m=3)

As versões 1, 2, 3 e 4 efectuaram sempre o mesmo número de iterações. A versão 5 não conseguiu convergir quando $\rho_{\max} = 5$.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	27	0e+00	0e+00	548	73	28	55	27	5	1
2	30	2e-07	9e-09	660	151	151	61	60	5	1
3	12	0e+00	0e+00	500	51	51	25	0	5	1
4	22	0e+00	0e+00	599	61	61	45	44	5	1
5	24	3e-07	1e-08	599	68	68	49	69	5	1
1	24	0e+00	0e+00	500	59	25	49	24	5	10
2	39	9e-11	4e-12	718	167	167	79	78	5	10
3	11	0e+00	0e+00	550	46	46	23	0	5	10
4	19	3e-15	0e+00	550	49	49	39	38	5	10
5	23	4e-07	2e-08	609	61	61	47	66	5	10
1	30	0e+00	0e+00	550	89	31	61	30	50	1
2	$> 10^3$	1e+05	3e-01						50	1
3	12	0e+00	0e+00	490	51	51	25	0	50	1
4	23	2e-14	1e-15	609	64	64	47	46	50	1
5	25	7e-07	3e-08	660	74	74	51	71	50	1
1	28	1e-13	2e-15	548	76	29	57	28	50	10
2	45	4e-07	2e-10	871	253	253	91	90	50	10
3	12	0e+00	0e+00	500	51	51	25	0	50	10
4	21	5e-14	1e-15	599	62	62	43	42	50	10
5	25	8e-07	4e-08	599	72	72	51	71	50	10
1	31	0e+00	0e+00	611	94	32	63	31	500	1
2	790	2e-09	1e-10	12410	7605	7605	1581	1580	500	1
3	13	0e+00	0e+00	548	57	57	27	0	500	1
4	28	0e+00	0e+00	650	86	86	57	56	500	1
5	30	9e-07	4e-08	611	76	76	61	86	500	1
1	30	0e+00	0e+00	599	89	31	61	30	500	10
2	415	5e-07	2e-08	6589	3843	3843	831	830	500	10
3	13	0e+00	0e+00	548	57	57	27	0	500	10
4	23	0e+00	0e+00	609	64	64	47	46	500	10
5	25	8e-07	4e-08	650	75	75	51	71	500	10

Tabela 6.8: Resultados do problema 6[†] [QQRT](n=2,m=1)

A versão 3 foi a que obteve melhores resultados ao nível do esforço computacional. As versões 1, 3 e 4 obtiveram os mesmos resultados em todas as tentativas. Pode-se ainda verificar que o tempo por iteração da versão 1 foi quase metade do tempo por iteração do da versão 3.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	8e-16	3e-01						5	1
2	28	8e-07	2e-07	830	217	217	57	112	5	1
3	$> 10^3$	9e-06	3e-12						5	1
4	21	5e-15	5e-15	830	201	201	43	84	5	1
5	39	6e-07	1e-08	1099	282	282	79	232	5	1
1	$> 10^3$	5e-15	6e-01						5	10
2	$> 10^3$	6e-13	3e-01						5	10
3	$> 10^3$	5e-06	5e-12						5	10
4	$> 10^3$	5e-16	2e+00						5	10
5	$> 10^3$	2e-10	2e+00						5	10
1	$> 10^3$	8e-16	1e-01						50	1
2	$> 10^3$	1e-15	2e-02						50	1
3	$> 10^3$	9e-06	3e-12						50	1
4	5	2e-15	1e-15	439	11	11	11	20	50	1
5	67	3e-07	2e-03	1759	631	631	135	398	50	1
1	$> 10^3$	1e-15	3e-01						50	10
2	211	4e-07	3e-07	3619	1223	1223	423	844	50	10
3	$> 10^3$	9e-06	3e-12						50	10
4	$> 10^3$	1e-01	3e-01						50	10
5	$> 10^3$	3e-01	1e-01						50	10
1	$> 10^3$	9e-15	4e-02						500	1
2	38	6e-08	2e-07	820	149	149	77	152	500	1
3	$> 10^3$	6e-06	8e-12						500	1
4	103	3e-09	3e-09	2628	958	958	207	412	500	1
5	26	8e-07	6e-07	830	175	175	53	154	500	1
1	$> 10^3$	9e-15	1e-01						500	10
2	202	1e-11	2e-12	3349	1055	1055	405	808	500	10
3	$> 10^3$	9e-06	3e-12						500	10
4	5	2e-16	4e-15	441	11	11	11	20	500	10
5	67	6e-07	4e-03	1759	627	627	135	398	500	10

Tabela 6.9: Resultados do problema 42[†] [QQTR](n=4,m=2)

A versões 1 e 3 não convergiram em nenhuma das tentativas. Foi a versão 4 que obteve o menor número de iterações (5), sendo nesses casos bastante mais rápida que qualquer tentativa com sucesso das outras versões.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	6e-14	7e-01						5	1
2	198	3e-07	4e-08	2910	940	940	397	792	5	1
3	$> 10^3$	8e-06	9e-12						5	1
4	$> 10^3$	3e-01	8e-01						5	1
5	$> 10^3$	3e-01	3e-01						5	1
1	$> 10^3$	2e-14	2e+00						5	10
2	$> 10^3$	1e-05	2e+00						5	10
3	$> 10^3$	8e-06	9e-12						5	10
4	$> 10^3$	1e-04	2e+00						5	10
5	$> 10^3$	2e-01	1e+00						5	10
1	$> 10^3$	3e-13	3e-01						50	1
2	$> 10^3$	1e+01	4e-01						50	1
3	$> 10^3$	3e-05	2e-11						50	1
4	$> 10^3$	7e-01	8e-02						50	1
5	$> 10^3$	1e-01	3e-02						50	1
1	$> 10^3$	7e-14	7e-01						50	10
2	$> 10^3$	1e+02	9e-01						50	10
3	$> 10^3$	8e-06	9e-12						50	10
4	$> 10^3$	6e-04	6e-01						50	10
5	$> 10^3$	2e+00	5e-01						50	10
1	$> 10^3$	8e-13	9e-02						500	1
2	75	2e-07	8e-09	1490	556	556	151	300	500	1
3	$> 10^3$	6e-05	6e-12						500	1
4	$> 10^3$	2e-01	2e-02						500	1
5	733	8e-07	3e-06	14939	6546	6546	1467	4342	500	1
1	$> 10^3$	9e-14	3e-01						500	10
2	505	3e-08	4e-09	8679	3809	3809	1011	2020	500	10
3	$> 10^3$	3e-05	2e-11						500	10
4	$> 10^3$	7e-01	8e-02						500	10
5	223	2e-07	1e-04	5000	1933	1933	447	1320	500	10

Tabela 6.10: Resultados do problema 61[†] [QQRT](n=3,m=2)

As versões 1, 3 e 4 não convergiram em nenhuma das tentativas. A versão 2 teve sucesso em três tentativas, obtendo numa delas o melhor resultado de todos (75 iterações). A versão 5 convergiu apenas duas vezes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	1e-16	1e-02						5	1
2	123	2e-08	2e-09	2199	1225	1225	247	246	5	1
3	$> 10^3$	1e+00	3e-01						5	1
4	OV									
5	$> 10^3$	2e+15	3e+07						5	1
1	$> 10^3$	2e-16	1e-02						5	10
2	$> 10^3$	1e-14	4e-01						5	10
3	$> 10^3$	1e+01	2e+00						5	10
4	OV									
5	$> 10^3$	2e+00	9e-02						5	10
1	$> 10^3$	1e-04	1e-02						50	1
2	OV									
3	$> 10^3$	1e-01	2e-02						50	1
4	17	1e-13	1e-14	550	38	38	35	34	50	1
5	39	4e-09	3e-09	769	165	165	79	115	50	1
1	$> 10^3$	4e-16	1e-02						50	10
2	$> 10^3$	6e-08	3e-01						50	10
3	$> 10^3$	3e-04	7e-06						50	10
4	52	1e-12	1e-13	1160	294	294	105	104	50	10
5	$> 10^3$	2e+06	4e+02						50	10
1	$> 10^3$	3e-03	2e-03						500	1
2	415	6e-08	8e-09	5599	2993	2993	831	830	500	1
3	$> 10^3$	5e-02	7e-03						500	1
4	19	1e-12	1e-13	548	61	61	39	38	500	1
5	70	1e-08	2e-08	1050	253	253	141	207	500	1
1	$> 10^3$	1e-04	8e-03						500	10
2	122	8e-08	1e-08	1970	1026	1026	245	244	500	10
3	$> 10^3$	3e-01	8e-02						500	10
4	36	1e-18	2e-34	929	330	330	73	72	500	10
5	104	2e-08	2e-08	1650	437	437	209	308	500	10

Tabela 6.11: Resultados do problema 27[†] [PQRT](n=3,m=1)

As versões 1 e 3 não convergiram em nenhuma das tentativas. A versão 4 teve sempre sucesso para $\rho_{\max} \in \{50, 500\}$, obtendo os melhores resultados. Tanto a versão 2 como a versão 5 convergiram três vezes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	48	3e-11	2e-12	160	109	49	97	48	5	1
2	57	4e-08	2e-12	500	179	179	115	114	5	1
3	18	8e-12	2e-12	270	50	50	37	0	5	1
4	40	9e-12	2e-12	709	83	83	81	80	5	1
5	28	2e-10	3e-11	549	60	60	57	82	5	1
1	47	3e-11	2e-12	660	108	48	95	47	5	10
2	31	4e-07	5e-08	549	94	94	63	62	5	10
3	18	8e-12	2e-12	549	50	50	37	0	5	10
4	38	1e-11	3e-12	659	78	78	77	76	5	10
5	28	3e-11	6e-12	609	59	59	57	82	5	10
1	51	3e-11	2e-12	650	119	52	103	51	50	1
2	57	2e-07	2e-11	770	191	191	115	114	50	1
3	18	8e-12	2e-12	549	50	50	37	0	50	1
4	42	7e-12	1e-12	709	87	87	85	84	50	1
5	28	2e-07	4e-08	610	85	85	57	82	50	1
1	47	3e-11	2e-12	599	108	48	95	47	50	10
2	57	6e-08	4e-12	770	165	165	115	114	50	10
3	18	8e-12	2e-12	549	50	50	37	0	50	10
4	39	8e-12	2e-12	719	80	80	79	78	50	10
5	28	2e-10	4e-11	599	59	59	57	82	50	10
1	51	3e-11	2e-12	599	137	52	103	51	500	1
2	94	7e-07	3e-08	1269	340	340	189	188	500	1
3	18	8e-12	2e-12	500	50	50	37	0	500	1
4	43	9e-12	2e-12	709	89	89	87	86	500	1
5	28	5e-09	1e-09	600	58	58	57	82	500	1
1	51	4e-11	3e-12	660	118	52	103	51	500	10
2	66	6e-07	1e-09	879	232	232	133	132	500	10
3	18	8e-12	2e-12	490	50	50	37	0	500	10
4	42	7e-12	2e-12	769	87	87	85	84	500	10
5	28	2e-08	5e-09	650	63	63	57	82	500	10

Tabela 6.12: Resultados do problema 26[†] [PPRT](n=3,m=1)

Houve convergência de todas as versões em todas as tentativas. A versão 3 foi a que efectuou menor número de iterações, necessitou de menor tempo de *CPU* e efectuou menor número de cálculos de f . Na versão 5, apesar de efectuar sempre o mesmo número de iterações, verificou-se que o número de cálculos de f , por exemplo, não foram os mesmos para todas as tentativas.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	1e-15	2e-01						5	1
2	15	5e-07	2e-07	610	40	40	31	90	5	1
3	$> 10^3$	1e-06	4e-12						5	1
4	4	5e-13	3e-13	440	9	9	9	24	5	1
5	10	1e-07	4e-08	609	30	30	21	87	5	1
1	$> 10^3$	8e-16	3e-01						5	10
2	15	7e-09	5e-09	549	39	39	31	90	5	10
3	$> 10^3$	1e-06	4e-12						5	10
4	4	2e-12	8e-13	440	9	9	9	24	5	10
5	10	2e-07	5e-08	549	24	24	21	87	5	10
1	$> 10^3$	2e-15	7e-02						50	1
2	20	1e-07	1e-07	660	55	55	41	120	50	1
3	$> 10^3$	1e-06	4e-12						50	1
4	5	3e-16	3e-17	439	11	11	11	30	50	1
5	10	2e-07	2e-08	549	25	25	21	87	50	1
1	$> 10^3$	2e-15	2e-01						50	10
2	14	7e-07	4e-07	610	38	38	29	84	50	10
3	$> 10^3$	1e-06	4e-12						50	10
4	4	4e-13	2e-13	440	9	9	9	24	50	10
5	10	1e-07	3e-08	549	27	27	21	87	50	10
1	$> 10^3$	4e-15	2e-02						500	1
2	36	7e-08	5e-08	940	124	124	73	216	500	1
3	$> 10^3$	1e-06	4e-12						500	1
4	5	1e-15	9e-16	440	11	11	11	30	500	1
5	9	1e-07	1e-08	549	19	19	19	78	500	1
1	$> 10^3$	3e-15	7e-02						500	10
2	21	3e-07	1e-07	660	57	57	43	126	500	10
3	$> 10^3$	1e-06	4e-12						500	10
4	5	2e-16	1e-16	489	11	11	11	30	500	10
5	10	2e-07	2e-08	550	25	25	21	87	500	10

Tabela 6.13: Resultados do problema 40[†] [PPRT](n=4,m=3)

Tanto a versão 1 como a versão 3 não convergiram em nenhuma das tentativas. Por outro lado, as versões 2, 4 e 5 convergiram em todas as tentativas. A versão 4 foi a que efectuou menor esforço computacional.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	2e-14	3e-01						5	1
2	OV									
3	32	7e-07	9e-14	979	165	165	65	0	5	1
4	$> 10^3$	3e+22	1e+13						5	1
5	$> 10^3$	5e+06	2e+03						5	1
1	$> 10^3$	7e-15	5e-01						5	10
2	OV									
3	56	7e-07	9e-14	1859	506	506	113	0	5	10
4	$> 10^3$	2e+12	4e+04						5	10
5	$> 10^3$	4e+16	8e+07						5	10
1	$> 10^3$	3e-14	1e-01						50	1
2	OV									
3	72	9e-07	1e-13	1919	480	480	145	0	50	1
4	40	2e-08	1e-12	1379	243	243	81	240	50	1
5	538	4e-07	7e-08	14939	4098	4098	1077	4827	50	1
1	$> 10^3$	6e-15	3e-01						50	10
2	OV									
3	31	7e-07	7e-14	940	157	157	63	0	50	10
4	$> 10^3$	1e+13	3e+04						50	10
5	110	9e-07	1e-06	3570	953	953	221	987	50	10
1	$> 10^3$	1e-13	4e-02						500	1
2	OV									
3	34	8e-07	1e-13	990	184	184	69	0	500	1
4	OV									
5	65	9e-07	1e-04	1759	349	349	131	582	500	1
1	$> 10^3$	1e-14	1e-01						500	10
2	OV									
3	52	7e-07	9e-14	1379	298	298	105	0	500	10
4	136	4e-08	2e-13	4830	1249	1249	273	816	500	10
5	45	9e-07	7e-08	1259	205	205	91	402	500	10

Tabela 6.14: Resultados do problema 47[†] [PPRT](n=5,m=3)

As versões 1 e 2 não convergiram em nenhuma das tentativas. No geral, a versão 3 foi a que permitiu obter melhores resultados. A versão 4 terminou apenas duas vezes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	2e-14	2e-01						5	1
2	34	4e-07	7e-08	870	123	123	69	204	5	1
3	$> 10^3$	3e-06	2e-12						5	1
4	8	3e-12	4e-13	500	17	17	17	48	5	1
5	11	2e-07	1e-08	550	23	23	23	96	5	1
1	$> 10^3$	4e-14	3e-01						5	10
2	27	8e-08	8e-09	820	123	123	55	162	5	10
3	$> 10^3$	3e-06	2e-12						5	10
4	8	2e-12	4e-13	500	17	17	17	48	5	10
5	11	1e-07	4e-09	549	23	23	23	96	5	10
1	$> 10^3$	2e-13	7e-02						50	1
2	56	6e-07	7e-08	1159	174	174	113	336	50	1
3	$> 10^3$	3e-06	2e-12						50	1
4	8	6e-15	5e-16	500	17	17	17	48	50	1
5	10	4e-07	4e-08	550	21	21	21	87	50	1
1	$> 10^3$	6e-14	2e-01						50	10
2	34	3e-07	2e-08	879	114	114	69	204	50	10
3	$> 10^3$	3e-06	2e-12						50	10
4	8	4e-12	8e-13	500	17	17	17	48	50	10
5	11	2e-07	2e-08	550	23	23	23	96	50	10
1	$> 10^3$	2e-13	3e-02						500	1
2	140	1e-06	9e-08	2580	479	479	281	840	500	1
3	$> 10^3$	3e-06	2e-12						500	1
4	8	1e-14	1e-15	500	17	17	17	48	500	1
5	11	6e-07	6e-08	539	23	23	23	96	500	1
1	$> 10^3$	8e-14	7e-02						500	10
2	177	3e-07	5e-08	3519	744	744	355	1062	500	10
3	$> 10^3$	3e-06	2e-12						500	10
4	8	6e-15	6e-16	490	17	17	17	48	500	10
5	9	4e-07	9e-08	549	19	19	19	78	500	10

Tabela 6.15: Resultados do problema 78[†] [PPRP](n=5,m=3)

Tanto a versão 1 como a versão 3 não convergiram em nenhuma das tentativas. Por outro lado, as versões 2, 4 e 5 tiveram sempre êxito. Podemos verificar que a versão 4 foi a que efectuou menos iterações em todas as tentativas.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	2e-14	1e-02						5	1
2	OV									
3	6	6e-09	3e-14	389	13	13	13	0	5	1
4	37	3e-11	6e-14	1319	254	254	75	222	5	1
5	134	9e-07	5e-07	3459	834	834	269	1200	5	1
1	$> 10^3$	8e-15	1e-02						5	10
2	OV									
3	6	6e-09	3e-14	439	13	13	13	0	5	10
4	OV									
5	19	6e-07	6e-07	819	94	94	39	168	5	10
1	$> 10^3$	8e-05	1e-02						50	1
2	OV									
3	6	6e-09	3e-14	440	13	13	13	0	50	1
4	OV									
5	19	8e-07	1e-08	659	54	54	39	168	50	1
1	$> 10^3$	9e-15	1e-02						50	10
2	OV									
3	6	6e-09	3e-14	439	13	13	13	0	50	10
4	14	6e-07	8e-07	600	30	30	29	84	50	10
5	260	9e-07	7e-07	7360	2028	2028	521	2331	50	10
1	$> 10^3$	4e-04	1e-02						500	1
2	250	9e-07	1e-07	8669	3010	3010	501	1500	500	1
3	6	6e-09	3e-14	440	13	13	13	0	500	1
4	11	6e-07	6e-07	549	24	24	23	66	500	1
5	25	1e-06	8e-07	770	67	67	51	222	500	1
1	$> 10^3$	1e-04	8e-03						500	10
2	OV									
3	6	6e-09	3e-14	440	13	13	13	0	500	10
4	10	7e-12	3e-16	549	31	31	21	60	500	10
5	43	1e-06	6e-05	1100	145	145	87	384	500	10

Tabela 6.16: Resultados do problema 79[†] [PPRP](n=5,m=3)

A versão 1 não terminou nenhuma das tentativas e a 2 terminou apenas uma das tentativas. A versão 5 terminou todas as tentativas. Foi com a versão 3 que se obtiveram os melhores resultados.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	35	1e-11	1e-12	540	73	36	71	70	5	1
2	126	7e-07	2e-07	1870	623	623	253	504	5	1
3	34	4e-12	1e-12	660	70	70	69	0	5	1
4	31	9e-12	2e-12	769	65	65	63	124	5	1
5	36	4e-11	1e-11	760	98	98	73	214	5	1
1	35	1e-11	1e-12	490	73	36	71	70	5	10
2	126	9e-07	2e-07	1870	630	630	253	504	5	10
3	34	4e-12	1e-12	660	70	70	69	0	5	10
4	31	9e-12	2e-12	759	68	68	63	124	5	10
5	32	1e-10	2e-11	770	81	81	65	190	5	10
1	35	1e-11	1e-12	500	73	36	71	70	50	1
2	138	8e-07	3e-07	1979	652	652	277	552	50	1
3	34	4e-12	1e-12	660	70	70	69	0	50	1
4	31	8e-12	2e-12	770	65	65	63	124	50	1
5	28	5e-10	5e-11	709	67	67	57	166	50	1
1	35	1e-11	1e-12	549	73	36	71	70	50	10
2	126	8e-07	2e-07	1810	631	631	253	504	50	10
3	34	4e-12	1e-12	660	70	70	69	0	50	10
4	33	8e-12	2e-12	820	75	75	67	132	50	10
5	33	9e-12	3e-12	769	85	85	67	196	50	10
1	41	9e-12	8e-13	550	108	42	83	82	500	1
2	135	4e-07	2e-07	1870	517	517	271	540	500	1
3	34	4e-12	1e-12	660	70	70	69	0	500	1
4	45	3e-14	1e-14	1040	160	160	91	180	500	1
5	36	2e-07	5e-08	830	149	149	73	214	500	1
1	35	1e-11	1e-12	549	73	36	71	70	500	10
2	126	8e-07	2e-07	1869	645	645	253	504	500	10
3	34	4e-12	1e-12	659	70	70	69	0	500	10
4	31	7e-12	2e-12	769	65	65	63	124	500	10
5	29	4e-10	5e-11	769	88	88	59	172	500	10

Tabela 6.17: Resultados do problema 46[†] [PGRT](n=5,m=2)

Verifica-se que todas as tentativas convergiram. As versões 1, 3, 4 e 5 terminaram com um número de iterações semelhante. No entanto, a versão 1 apresentou o melhor desempenho ao nível do tempo de *CPU*. A versão 2 é de todas a que obteve piores resultados.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	6e-15	3e-01						5	1
2	OV									
3	36	1e-07	1e-07	1209	190	190	73	0	5	1
4	$> 10^3$	3e-06	4e-01						5	1
5	186	9e-12	3e-12	8400	1820	1820	373	2228	5	1
1	$> 10^3$	6e-15	7e-01						5	10
2	OV									
3	31	2e-07	1e-07	979	109	109	63	0	5	10
4	46	7e-12	3e-12	2530	464	464	93	368	5	10
5	519	1e-11	1e-06	19390	3564	3564	1039	6220	5	10
1	$> 10^3$	1e-14	1e-01						50	1
2	347	8e-07	5e-07	8300	953	953	695	2776	50	1
3	68	1e-07	1e-07	2850	721	721	137	0	50	1
4	15	3e-14	2e-14	1040	151	151	31	120	50	1
5	23	3e-07	8e-08	1160	159	159	47	272	50	1
1	$> 10^3$	4e-15	3e-01						50	10
2	OV									
3	29	1e-07	1e-07	929	101	101	59	0	50	10
4	$> 10^3$	3e-06	3e-01						50	10
5	719	3e-10	7e-06	27849	4946	4946	1439	8616	50	10
1	$> 10^3$	7e-14	4e-02						500	1
2	OV									
3	124	4e-07	1e-07	5219	1392	1392	249	0	500	1
4	15	2e-15	2e-16	930	74	74	31	120	500	1
5	23	4e-07	1e-07	1209	158	158	47	272	500	1
1	$> 10^3$	1e-14	1e-01						500	10
2	186	1e-08	5e-09	6540	1312	1312	373	1488	500	10
3	69	2e-07	1e-07	2849	716	716	139	0	500	10
4	29	2e-15	3e-16	1860	369	369	59	232	500	10
5	21	8e-07	2e-07	1150	167	167	43	248	500	10

Tabela 6.18: Resultados do problema 56[†] [PGRT](n=7,m=4)

A versão 1 não convergiu em nenhuma das tentativas. A versão 2 convergiu apenas duas vezes, requerendo, em comparação com as versões 3, 4 e 5, muito maior esforço computacional. Ao nível do número de iterações foi a versão 4 que obteve o melhor desempenho.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	3e-13	3e-02						5	1
2	43	5e-07	2e-08	830	135	135	87	172	5	1
3	12	2e-07	8e-13	439	27	27	25	0	5	1
4	15	8e-11	2e-12	600	34	34	31	60	5	1
5	412	7e-08	2e-09	9719	4048	4048	825	2464	5	1
1	$> 10^3$	2e-13	3e-02						5	10
2	44	4e-07	2e-08	820	190	190	89	176	5	10
3	12	2e-07	8e-13	500	27	27	25	0	5	10
4	OV									
5	396	2e-07	3e-09	9070	3843	3843	793	2368	5	10
1	$> 10^3$	1e-13	2e-02						50	1
2	59	6e-07	1e-07	929	188	188	119	236	50	1
3	12	2e-07	8e-13	500	27	27	25	0	50	1
4	16	2e-14	9e-16	550	51	51	33	64	50	1
5	$> 10^3$	2e+05	3e+01						50	1
1	$> 10^3$	2e-13	3e-02						50	10
2	OV									
3	12	2e-07	8e-13	500	27	27	25	0	50	10
4	12	2e-14	9e-16	549	27	27	25	48	50	10
5	$> 10^3$	2e+05	3e+01						50	10
1	$> 10^3$	3e-03	3e-02						500	1
2	OV									
3	12	2e-07	8e-13	439	27	27	25	0	500	1
4	15	4e-14	2e-15	610	33	33	31	60	500	1
5	17	4e-08	4e-10	549	37	37	35	100	500	1
1	$> 10^3$	2e-13	2e-02						500	10
2	OV									
3	12	2e-07	8e-13	440	27	27	25	0	500	10
4	16	1e-13	4e-15	609	39	39	33	64	500	10
5	16	1e-07	3e-09	549	34	34	33	94	500	10

Tabela 6.19: Resultados do problema 77[†] [PGRP](n=5,m=2)

A versão 1 não convergiu em nenhuma das tentativas. Os melhores desempenhos, tanto no número de iterações (à exceção de um caso com a versão 4) como no tempo de *CPU*, foi obtido com a versão 3.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	3e-15	1e-02						5	1
2	15	3e-13	6e-14	490	65	65	31	30	5	1
3	5	4e-17	0e+00	440	11	11	11	0	5	1
4	5	2e-15	4e-16	439	11	11	11	10	5	1
5	16	3e-08	6e-09	490	42	42	33	46	5	1
1	$> 10^3$	1e-14	1e-02						5	10
2	21	2e-07	3e-08	500	66	66	43	42	5	10
3	5	4e-17	0e+00	440	11	11	11	0	5	10
4	5	2e-15	4e-16	389	11	11	11	10	5	10
5	14	1e-07	2e-08	489	30	30	29	40	5	10
1	$> 10^3$	1e-08	8e-03						50	1
2	17	2e-07	5e-08	500	74	74	35	34	50	1
3	5	4e-17	0e+00	380	11	11	11	0	50	1
4	5	2e-15	4e-16	439	11	11	11	10	50	1
5	15	1e-08	3e-09	490	48	48	31	43	50	1
1	$> 10^3$	2e-15	1e-02						50	10
2	20	1e-08	2e-12	489	80	80	41	40	50	10
3	5	4e-17	0e+00	390	11	11	11	0	50	10
4	5	2e-15	4e-16	439	11	11	11	10	50	10
5	15	4e-09	7e-10	500	39	39	31	43	50	10
1	$> 10^3$	6e-08	2e-03						500	1
2	16	5e-10	1e-10	490	91	91	33	32	500	1
3	5	4e-17	0e+00	440	11	11	11	0	500	1
4	5	2e-15	4e-16	439	11	11	11	10	500	1
5	17	1e-07	3e-08	500	47	47	35	48	500	1
1	$> 10^3$	1e-08	8e-03						500	10
2	13	1e-10	2e-11	429	77	77	27	26	500	10
3	5	4e-17	0e+00	440	11	11	11	0	500	10
4	5	2e-15	4e-16	389	11	11	11	10	500	10
5	21	9e-07	2e-07	540	54	54	43	60	500	10

Tabela 6.20: Resultados do problema 9^\dagger [GLRT](n=2,m=1)

A versão 1 não convergiu em nenhuma das tentativas. As versões 3 e 4 tiveram sempre o mesmo desempenho ao longo das seis tentativas. As versões 2 e 5 tiveram em quase todas as tentativas desempenhos semelhantes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	1e-15	5e-01						5	1
2	33	6e-08	3e-08	989	255	255	67	198	5	1
3	2	4e-16	6e-16	390	5	5	5	0	5	1
4	$> 10^3$	2e+02	7e+02						5	1
5	$> 10^3$	2e+01	3e+00						5	1
1	$> 10^3$	1e-15	7e-01						5	10
2	174	7e-07	3e-07	5219	2619	2619	349	1044	5	10
3	2	4e-16	6e-16	430	5	5	5	0	5	10
4	$> 10^3$	2e+02	7e+02						5	10
5	$> 10^3$	1e+01	4e+00						5	10
1	$> 10^3$	1e-15	2e-01						50	1
2	748	1e-06	5e-07	13840	4839	4839	1497	4488	50	1
3	2	4e-16	6e-16	389	5	5	5	0	50	1
4	$> 10^3$	1e-01	5e-02						50	1
5	$> 10^3$	4e-02	1e-02						50	1
1	$> 10^3$	4e-15	4e-01						50	10
2	242	1e-06	2e-07	5599	2260	2260	485	1452	50	10
3	2	4e-16	6e-16	390	5	5	5	0	50	10
4	$> 10^3$	1e+00	2e+00						50	10
5	$> 10^3$	4e-01	1e-01						50	10
1	$> 10^3$	1e-14	6e-02						500	1
2	133	5e-07	8e-10	2250	473	473	267	798	500	1
3	2	4e-16	6e-16	389	5	5	5	0	500	1
4	6	2e-15	7e-16	500	15	15	13	36	500	1
5	142	7e-07	3e-05	3570	1222	1222	285	1272	500	1
1	$> 10^3$	2e-15	2e-01						500	10
2	98	2e-07	2e-08	2199	684	684	197	588	500	10
3	2	4e-16	6e-16	380	5	5	5	0	500	10
4	6	1e-15	5e-16	489	15	15	13	36	500	10
5	$> 10^3$	2e-02	7e-03						500	10

Tabela 6.21: Resultados do problema 53[†] [QLRT](n=5,m=3)

A versão 1 não convergiu em nenhuma das tentativas. A versão 3 foi a que obteve melhores resultados, tendo em todas as tentativas o mesmo custo computacional. A versão 2 também convergiu em todas as tentativas. A versão 5 terminou apenas uma vez.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	10	2e-14	4e-03	430	25	11	21	10	5	1
2	40	1e-07	8e-09	609	193	193	81	80	5	1
3	8	2e-09	1e-15	439	17	17	17	0	5	1
4	8	6e-16	4e-17	440	19	19	17	16	5	1
5	9	1e-09	2e-09	439	26	26	19	26	5	1
1	7	2e-12	4e-03	390	16	8	15	7	5	10
2	21	7e-09	3e-10	549	74	74	43	42	5	10
3	8	2e-09	1e-15	389	17	17	17	0	5	10
4	7	6e-16	4e-17	440	16	16	15	14	5	10
5	12	2e-10	4e-12	500	46	46	25	35	5	10
1	11	2e-14	4e-03	439	25	12	23	11	50	1
2	OV									
3	8	2e-09	1e-15	439	17	17	17	0	50	1
4	34	4e-14	3e-15	660	121	121	69	68	50	1
5	10	4e-10	1e-09	500	42	42	21	29	50	1
1	8	9e-11	4e-03	440	18	9	17	8	50	10
2	89	3e-07	2e-08	1160	753	753	179	178	50	10
3	8	2e-09	1e-15	439	17	17	17	0	50	10
4	7	6e-16	4e-17	440	18	18	15	14	50	10
5	10	1e-10	8e-11	439	27	27	21	29	50	10
1	10	2e-14	4e-03	439	22	11	21	10	500	1
2	69	3e-07	2e-08	820	357	357	139	138	500	1
3	8	2e-09	1e-15	379	17	17	17	0	500	1
4	7	7e-11	5e-12	440	16	16	15	14	500	1
5	10	7e-09	3e-10	500	22	22	21	29	500	1
1	11	2e-14	4e-03	389	25	12	23	11	500	10
2	OV									
3	8	2e-09	1e-15	439	17	17	17	0	500	10
4	9	8e-15	4e-16	439	25	25	19	18	500	10
5	10	3e-10	1e-10	490	27	27	21	29	500	10

Tabela 6.22: Resultados do problema 26^{††}[PPRT](n=3,m=1)

A versão 1, ao terminar todas as tentativas, obteve desempenhos semelhantes às versões 3 e 4, relativamente ao número de iterações e custos de *CPU*. A versão 4 foi a que conseguiu convergir no menor número de iterações, o que não foi traduzido nos custos computacionais.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	7e-14	5e-01						5	1
2	$> 10^3$	8e-13	2e+00						5	1
3	184	1e-06	8e-15	3180	686	686	369	0	5	1
4	OV									
5	$> 10^3$	9e+60	7e+35						5	1
1	$> 10^3$	8e-15	1e+00						5	10
2	OV									
3	57	3e-07	2e-14	1419	381	381	115	0	5	10
4	OV									
5	$> 10^3$	5e+09	7e+05						5	10
1	$> 10^3$	1e-13	4e-01						50	1
2	133	1e-08	1e-08	2309	514	514	267	798	50	1
3	68	7e-07	1e-13	1700	392	392	137	0	50	1
4	$> 10^3$	4e+00	5e-01						50	1
5	78	4e-07	2e-08	2139	526	526	157	699	50	1
1	$> 10^3$	3e-14	8e-01						50	10
2	$> 10^3$	2e-12	1e+00						50	10
3	27	4e-07	4e-14	769	103	103	55	0	50	10
4	$> 10^3$	1e-03	3e+00						50	10
5	OV									
1	$> 10^3$	6e-13	6e-02						500	1
2	OV									
3	$> 10^3$	2e-04	4e-10						500	1
4	OV									
5	36	4e-07	6e-09	1050	140	140	73	321	500	1
1	$> 10^3$	5e-14	4e-01						500	10
2	159	5e-07	2e-07	3189	1004	1004	319	954	500	10
3	48	3e-07	3e-14	1309	277	277	97	0	500	10
4	OV									
5	41	9e-07	1e-08	1380	266	266	83	366	500	10

Tabela 6.23: Resultados do problema 79^{††}[PPRT](n=5,m=3)

A versão 1 não convergiu em nenhuma das tentativas. A versão 3 foi a que permitiu resolver o problema com o menor número de iterações e custos computacionais para $\rho_{\max} = 50$ e $\Delta = 10$.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	5e-14	2e-02						5	1
2	OV									
3	12	2e-08	4e-14	440	26	26	25	0	5	1
4	18	4e-14	1e-15	610	41	41	37	72	5	1
5	$> 10^3$	4e+03	2e+01						5	1
1	$> 10^3$	1e-13	2e-02						5	10
2	OV									
3	12	2e-08	4e-14	490	26	26	25	0	5	10
4	14	1e-12	3e-14	549	32	32	29	56	5	10
5	$> 10^3$	4e+03	2e+01						5	10
1	$> 10^3$	9e-14	1e-02						50	1
2	49	8e-07	3e-08	830	184	184	99	196	50	1
3	12	2e-08	4e-14	439	26	26	25	0	50	1
4	23	9e-13	5e-14	709	60	60	47	92	50	1
5	163	1e-06	4e-08	3740	1365	1365	327	974	50	1
1	$> 10^3$	2e-13	2e-02						50	10
2	OV									
3	12	2e-08	4e-14	439	26	26	25	0	50	10
4	16	5e-14	2e-15	660	56	56	33	64	50	10
5	170	9e-08	2e-09	3850	1405	1405	341	1016	50	10
1	$> 10^3$	3e-03	2e-02						500	1
2	66	8e-07	3e-08	990	214	214	133	264	500	1
3	12	2e-08	4e-14	490	26	26	25	0	500	1
4	18	4e-14	1e-15	599	40	40	37	72	500	1
5	24	9e-07	1e-07	660	57	57	49	142	500	1
1	$> 10^3$	8e-14	1e-02						500	10
2	OV									
3	12	2e-08	4e-14	490	26	26	25	0	500	10
4	11	8e-13	4e-14	490	23	23	23	44	500	10
5	14	1e-08	2e-10	549	33	33	29	82	500	10

Tabela 6.24: Resultados do problema 77^{††}[PGRT](n=5,m=2)

A versão 1 não convergiu em nenhuma das tentativas. No geral, a versão 3 foi a que teve melhor desempenho, efectuando o mesmo esforço computacional em todas as tentativas.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	OV									
2	OV									
3	OV									
4	OV									
5	OV									
1	$> 10^3$	3e-14	2e-02						5	10
2	OV									
3	OV									
4	OV									
5	OV									
1	$> 10^3$	5e-14	4e-02						50	1
2	OV									
3	OV									
4	OV									
5	151	6e-07	2e-07	5000	1514	1514	303	1353	50	1
1	$> 10^3$	3e-14	2e-02						50	10
2	OV									
3	OV									
4	OV									
5	194	4e-08	3e-08	6750	2122	2122	389	1740	50	10
1	521	2e-08	1e-09	4119	1530	522	1043	1563	500	1
2	OV									
3	OV									
4	137	2e-14	3e-15	3679	954	954	275	822	500	1
5	148	2e-07	4e-08	3519	696	696	297	1326	500	1
1	$> 10^3$	2e-04	2e-02						500	10
2	OV									
3	OV									
4	OV									
5	OV									

Tabela 6.25: Resultados do problema 80[†][GPRP](n=5,m=3)

As versões 2 e 3 não convergiram em nenhuma das tentativas. As versões 1 e 4 convergiram apenas uma vez e a versão 5 convergiu três vezes.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	1e-14	2e-02						5	1
2	OV									
3	173	1e-10	3e-11	8780	2809	2809	347	0	5	1
4	8	1e-14	5e-16	500	17	17	17	48	5	1
5	27	2e-07	2e-07	929	107	107	55	240	5	1
1	$> 10^3$	1e-14	2e-02						5	10
2	OV									
3	175	3e-10	9e-11	8779	2832	2832	351	0	5	10
4	8	2e-13	3e-14	550	17	17	17	48	5	10
5	24	9e-07	2e-07	929	141	141	49	213	5	10
1	$> 10^3$	5e-04	1e-02						50	1
2	OV									
3	177	3e-10	7e-11	8839	2846	2846	355	0	50	1
4	8	2e-14	9e-16	549	17	17	17	48	50	1
5	14	8e-07	1e-08	600	36	36	29	123	50	1
1	$> 10^3$	1e-14	2e-02						50	10
2	OV									
3	175	1e-10	4e-11	8839	2833	2833	351	0	50	10
4	8	3e-14	1e-15	490	17	17	17	48	50	10
5	36	8e-07	7e-08	1160	156	156	73	321	50	10
1	$> 10^3$	3e-02	1e-03						500	1
2	OV									
3	172	1e-10	3e-11	8729	2797	2797	345	0	500	1
4	8	1e-12	1e-13	550	17	17	17	48	500	1
5	16	6e-07	5e-09	660	43	43	33	141	500	1
1	$> 10^3$	5e-04	1e-02						500	10
2	OV									
3	179	3e-10	7e-11	8889	2853	2853	359	0	500	10
4	8	2e-14	9e-16	549	17	17	17	48	500	10
5	16	2e-07	4e-09	660	45	45	33	141	500	10

Tabela 6.26: Resultados do problema 81[†][GPRP](n=5,m=3)

As versões 1 e 2 não convergiram em nenhuma das tentativas. Por outro lado, as outras versões convergiram nas seis tentativas. O melhor desempenho foi obtido pela versão 4 (número de iterações e tempos de *CPU*).

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	4e-14	2e-01						5	1
2	32	2e-07	4e-09	601	116	116	65	128	5	1
3	215	6e-07	2e-10	4830	2408	2408	431	0	5	1
4	8	6e-12	6e-13	500	43	43	17	32	5	1
5	9	1e-07	2e-08	490	42	42	19	52	5	1
1	$> 10^3$	9e-14	4e-01						5	10
2	34	9e-08	5e-09	660	122	122	69	136	5	10
3	131	6e-07	2e-10	2250	875	875	263	0	5	10
4	8	2e-13	2e-14	490	43	43	17	32	5	10
5	9	6e-07	1e-08	441	42	42	19	52	5	10
1	$> 10^3$	1e-13	8e-02						50	1
2	98	4e-07	2e-08	1429	567	567	197	392	50	1
3	$> 10^3$	3e+00	4e-02						50	1
4	8	3e-14	2e-15	490	42	42	17	32	50	1
5	10	1e-07	8e-09	500	46	46	21	58	50	1
1	$> 10^3$	4e-13	2e-01						50	10
2	26	3e-07	3e-08	609	92	92	53	104	50	10
3	$> 10^3$	8e+00	4e-01						50	10
4	8	3e-14	2e-15	429	42	42	17	32	50	10
5	10	5e-08	4e-09	500	46	46	21	58	50	10
1	$> 10^3$	2e-13	3e-02						500	1
2	179	6e-07	3e-08	2308	1086	1086	359	716	500	1
3	$> 10^3$	3e+00	3e-02						500	1
4	9	2e-14	1e-15	500	44	44	19	36	500	1
5	9	8e-08	3e-09	490	44	44	19	52	500	1
1	$> 10^3$	5e-13	8e-02						500	10
2	168	5e-07	2e-08	2310	1179	1179	337	672	500	10
3	$> 10^3$	2e+00	6e-02						500	10
4	8	4e-14	2e-15	439	42	42	17	32	500	10
5	8	6e-07	5e-08	488	42	42	17	46	500	10

Tabela 6.27: Resultados do problema $63^\dagger[\text{QQR}P](n=3,m=2)$

A versão 1 não convergiu em nenhuma das tentativas. As versões 4 e 5 tiveram desempenhos semelhantes. Verificou-se ainda que a versão 2 convergiu nas seis tentativas.

V	Nit	$\ \nabla\Psi\ _2$	$\ c\ _2$	Tmp	N_f	$N_{\nabla f}$	$N_{\nabla^2 f}$	$N_{\nabla^2 c_i}$	ρ_{\max}	Δ
1	$> 10^3$	7e+17	3e-01						5	1
2	OV									
3	$> 10^3$	3e+23	3e+04						5	1
4	$> 10^3$	5e+28	4e+13						5	1
5	$> 10^3$	2e+52	1e+26						5	1
1	$> 10^3$	7e+17	3e-01						5	10
2	OV									
3	OV									
4	$> 10^3$	2e+25	1e+11						5	10
5	$> 10^3$	1e+37	3e+18						5	10
1	$> 10^3$	7e+17	3e-01						50	1
2	OV									
3	$> 10^3$	2e+20	7e+01						50	1
4	$> 10^3$	2e+22	3e+10						50	1
5	OV									
1	$> 10^3$	7e+17	3e-01						50	10
2	OV									
3	$> 10^3$	2e+20	7e+01						50	10
4	$> 10^3$	6e+27	1e+12						50	10
5	OV									
1	$> 10^3$	7e+17	3e-01						500	1
2	51	2e-08	1e-09	818	747	747	103	102	500	1
3	$> 10^3$	2e+20	7e+01						500	1
4	$> 10^3$	4e+19	4e+07						500	1
5	OV									
1	$> 10^3$	7e+17	3e-01						500	10
2	OV									
3	$> 10^3$	3e+23	3e+04						500	10
4	$> 10^3$	7e+31	1e+15						500	10
5	74	4e-12	0e+00	1210	782	782	149	220	500	10

Tabela 6.28: Resultados do problema $41^+[\text{PLRT}](n=4,m=1)$

As versões 1, 3 e 4 não convergiram em nenhuma das tentativas. Tanto a versão 2 como a versão 5 convergiram apenas uma vez.

Da análise estatística das tabelas 6.2 a 6.28, e considerando apenas as tentativas em que existiu convergência, podemos retirar os seguintes resultados para os totais do número de iterações realizadas, dos cálculos de f^2 e do tempo gasto (em milésimos de segundo).

	Versão 1	Versão 2	Versão 3	Versão 4	Versão 5
Iterações	1.571	11.216	3.388	2.143	7.960
Cálculos de f	3.967	73.109	29.410	9.244	56.443
Tempo Gasto	29.162	223.805	131.953	81.511	233.946

Tabela 6.29: Custos computacionais totais (considerando apenas as tentativas em que existiu convergência).

Das Tabelas 6.1 e 6.29, obtemos a seguinte tabela onde estão indicados os valores médios do número de iterações realizadas, dos cálculos de f e do tempo gasto.

	Versão 1	Versão 2	Versão 3	Versão 4	Versão 5
Iterações	28,6	103,9	29,5	18,0	67,5
Cálculos de f	72,1	676,9	255,7	77,7	478,3
Tempo de <i>CPU</i>	530	2.072	1.147	685	1.982

Tabela 6.30: Média do número de iterações, cálculos da função e tempo de *CPU* (considerando apenas as tentativas em que existiu convergência).

A versão 1, como se pode verificar das tabelas (6.2 a 6.28), convergiu apenas nos problemas em que, geralmente, também todos os outros convergiram (os mais simples). Assim, considerando apenas as versões mais robustas (versões 2, 3, 4 e 5) verificamos, da Tabela 6.30, que a versão 4 foi a que obteve em média os melhores resultados para os parâmetros estudados.

A seguir à versão 4, a versão que obteve melhores resultados foi a versão 3. No entanto, nota-se uma diferença bastante significativa para a primeira pois, realizou em média mais do triplo dos cálculos da função f e necessitou de 1,7 vezes mais tempo de *CPU*. Tanto a versão 2 como a versão 5, necessitaram, em média, de aproximadamente o triplo do tempo de *CPU* requerido pela versão 4, sendo a razão superior ao comparar o número de cálculos da função.

²Para qualquer das versões, o número de cálculos de f é igual ao número de cálculos de Ψ .

Relativamente à versão 2, esta é a versão com menor tempo de *CPU* por iteração (veja-se a Tabela 6.31). No entanto, e visto o seu número médio de iterações ser 103,9, mais que o quántuplo do número de iterações médias da versão 4, o tempo de *CPU* gasto foi, em média, superior às outras versões. Pela mesma razão, também o número médio de cálculos de f foi bastante superior à média das outras versões.

	Versão 1	Versão 2	Versão 3	Versão 4	Versão 5
Tempo médio por iteração	18,6	20,0	38,9	38,0	29,4

Tabela 6.31: Tempo médio de *CPU* por iteração.

Como se pode verificar das tabelas de resultados e da Tabela 6.31, a versão 1 foi a que apresentou a melhor razão *tempo por iteração*. Tendo em conta os resultados, principalmente das versões 2 e 5 e como até certo ponto seria de esperar, parece obter-se alguma vantagem, em termos de requerimentos de *CPU*, ao utilizar as fórmulas Quasi-Newton.

Agora, considerando somente o número de iterações efectuadas, verifica-se que a versão 3 foi a que no total das 162 tentativas por mais vezes (55,6%) efectuou menos iterações (veja-se a Tabela 6.32). No entanto, considerando os 27 problemas, a versão 4 foi a que, para alguma combinação dos parâmetros (ρ_{\max} , Δ), conseguiu efectuar o menor número de iterações mais vezes (59,3%). Nestes termos, os piores resultados foram obtidos pelas versões 2 e 5.

	Versão 1	Versão 2	Versão 3	Versão 4	Versão 5
27 problemas	14,8%	7,4%	48,1%	59,3%	11,1%
162 tentativas	13,6%	3,1%	55,6%	44,4%	9,3%

Tabela 6.32: Percentagem de vezes que realizaram menor ou igual número de iterações.

Relativamente à escolha do parâmetro de penalização (Algoritmo 6.2), verificou-se que foi para a escolha de $\rho_{\max} = 500$ e $\Delta = 1$ que se obtiveram os melhores resultados ao nível da robustez para as versões 1, 2, 4 e 5 (vejam-se as Tabelas 6.33 e 6.34). A versão 3 obteve resultados semelhantes para as várias combinações, com uma pequena vantagem em considerar $\rho_{\max} = 5$.

	(ρ_{\max}, Δ)					
	(5,1)	(5,10)	(50,1)	(50,10)	(500,1)	(500,10)
Versão 1	9	9	9	9	10	9
Versão 2	19	16	17	16	21	19
Versão 3	20	20	19	19	18	19
Versão 4	18	16	21	18	23	23
Versão 5	16	15	21	18	26	22
	82	76	87	80	98	92

Tabela 6.33: Número de sucessos para cada escolha dos parâmetros (ρ_{\max}, Δ) , no total das 135 tentativas.

	(ρ_{\max}, Δ)					
	(5,1)	(5,10)	(50,1)	(50,10)	(500,1)	(500,10)
Terminou nas 135 tentativas(%)	60,7	56,2	64,4	59,2	72,6	68,1

Tabela 6.34: Percentagem de sucessos para cada escolha dos parâmetros ρ_{\max} e Δ .

Finalmente, quanto à relação entre custos computacionais e escolha de ρ_{\max} e Δ , verificou-se que a escolha mais robusta ($\rho_{\max} = 500, \Delta = 1$) foi a que efectuou, em média, maior número de iterações (ver Tabela 6.36). Este facto, estará relacionado com a não resolução dos problemas mais exigentes pelas outras escolhas dos parâmetros. A corroborar este facto, temos que a escolha menos robusta (com 56,2% de sucessos) foi a que obteve menor número total de iterações (ver Tabela 6.35).

6.5 Síntese

As cinco versões implementadas para a resolução de problemas com restrições de igualdade, numa filosofia de técnica de penalização exacta, são:

1. Método de segurança de Newton com procura unidimensional na resolução do problema sem restrições, baseado na fórmula de substituição que é apenas função das restrições;

	(ρ_{\max}, Δ)					
	(5,1)	(5,10)	(50,1)	(50,10)	(500,1)	(500,10)
Versão 1	172	165	179	170	706	179
Versão 2	1405	774	2056	1227	2771	2983
Versão 3	802	611	547	424	493	511
Versão 4	281	246	344	284	541	447
Versão 5	1004	1173	1284	1711	1945	843
Total de iterações	3664	2969	4410	3816	6456	4963

Tabela 6.35: Número de iterações para cada escolha dos parâmetros ρ_{\max} e Δ .

	(ρ_{\max}, Δ)					
	(5,1)	(5,10)	(50,1)	(50,10)	(500,1)	(500,10)
Média de iterações por sucesso	44,7	39,1	53,8	47,1	65,9	53,9

Tabela 6.36: Média de iterações por sucesso para cada escolha dos parâmetros ρ_{\max} e Δ .

2. Método Quasi-Newton baseado na fórmula de actualização de BFGS com procura unidimensional na resolução do problema sem restrições, em que a fórmula de substituição do vector dos multiplicadores é função do gradiente da função objectivo e da inversa generalizada do Jacobiano das restrições;
3. Método Pseudo-Newton com procura unidimensional na resolução do problema sem restrições, em que a fórmula de substituição do vector dos multiplicadores é função do gradiente da função objectivo e da inversa generalizada do Jacobiano das restrições;
4. Método de segurança de Newton com procura unidimensional na resolução do problema sem restrições, em que é usada a fórmula de actualização Quasi-Newton BFGS para aproximar as Hessianas do vector dos multiplicadores de Lagrange, e a fórmula de substituição do vector dos multiplicadores é função do gradiente da função objectivo e da inversa generalizada do Jacobiano das restrições;

5. Método de segurança de Newton com procura unidimensional na resolução do problema sem restrições, em que são usadas as fórmulas de actualização Quasi-Newton de Broyden, para aproximar a matriz do Jacobiano, e a de BFGS para aproximar as Hessianas, do vector dos multiplicadores de Lagrange, e a fórmula de substituição do vector dos multiplicadores é função do gradiente da função objectivo e da inversa generalizada do Jacobiano das restrições.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

De acordo com o estudo teórico realizado no Capítulo 5, para a resolução de um problema de optimização não linear com restrições do tipo igualdade, decidiu-se recorrer a uma técnica de penalização exacta baseada na substituição do vector dos multiplicadores de Lagrange na função Lagrangeana aumentada. Desta forma, obtém-se uma função auxiliar sem restrições, cujo minimizante coincide com o minimizante do problema original e para o qual existe um leque de métodos para o calcular. Tendo-se optado pelo método de Newton, e ao recorrer às fórmulas de substituição do vector dos multiplicadores de Lagrange descritas nas secções 5.3.1 e 5.3.2, verificou-se que para a segunda fórmula a implementação do método de Newton seria virtualmente impossível devido à complexidade das expressões obtidas. Para fazer face a esta situação, optou-se por implementar quatro variantes do método de Newton, com o objectivo de comparar a eficiência e a robustez. As versões testadas foram:

- Uma versão Newton utilizando a primeira fórmula de substituição;

E utilizando a segunda fórmula de substituição:

- Uma versão Quasi-Newton em que a inversa da Hessiana da Lagrangeana aumentada é aproximada utilizando a fórmula de actualização BFGS;
- uma versão Newton em que se recorre ao uso de Pseudo-derivadas para

aproximar o Jacobiano do vector dos multiplicadores e a Hessina da função Lagrangeana aumentada;

- Uma versão mista Newton/Quasi-Newton em que se recorre ao uso da fórmula de actualização BFGS para Hessianas, para aproximar as Hessianas do vector dos multiplicadores de Lagrange e formar o sistema Newton;
- Uma outra versão mista Newton/Quasi-Newton em que se recorre não só ao uso da fórmula de actualização BFGS conforme a versão anterior, mas também à fórmula de actualização de Broyden para aproximar o Jacobiano do vector dos multiplicadores de Lagrange, conseguindo-se deste modo formar o sistema Newton.

Das execuções para o conjunto de problemas citado, verificou-se que existem diferenças significativas entre algumas das versões. Assim, verificámos o seguinte:

Versão 1

Esta versão é pouco robusta. Analisando as tabelas, verifica-se que na maioria dos testes o processo iterativo converge para um ponto onde o gradiente da Lagrangeana aumentada é aproximadamente zero. Esse ponto não é, em geral, um ponto admissível, sendo um ponto onde as parcelas $\nabla f(x)$ e $3\rho\nabla c(x)^T c(x)$, cuja soma é $\nabla\Psi_1$, são quase simétricas. Assim, observa-se também que, em geral, nos problemas em que o processo iterativo converge, o ponto que verifica o critério de paragem é um ponto crítico de f e também ponto admissível ou está muito próximo da região admissível (por forma a satisfazer no critério de convergência $\|c(x)\|_2 \leq 10^{-2}$). Genericamente, das vezes que convergiu apresentou resultados mais ou menos semelhantes com as versões 3 e 4.

Versão 2

Ao resolver 88,9% dos problemas testados, esta versão poderá considerar-se robusta. No entanto, verificamos que apenas resolve 66,7% do total das 162 tentativas. Esta versão efectua em média muito mais iterações que qualquer uma das outras. Verifica-se, no entanto, como seria de esperar, que o tempo médio por iteração é

o menor entre as versões que usam a segunda fórmula de substituição. Nota-se ainda que obteve a maior robustez quando se considerou $\rho_{\max} = 500$ e $\Delta = 1$ sendo, no entanto, para estes valores que efectua o segundo maior número total (e médio com 132 iterações por sucesso) de iterações.

Versão 3

Conseguindo resolver 71% das 162 tentativas, esta versão obteve um grau de robustez semelhante ao das versões 4 e 5, respectivamente com 73,5% e 72,8%. Por outro lado, como seria de esperar ao ter de calcular analiticamente expressões relativamente complexas, esta é a versão com maior razão tempo/iteração. Nota-se ainda que obteve a maior robustez quando se considerou $\rho_{\max} = 5$, sendo no entanto para este valor que efectua o maior número de iterações.

Versão 4

Esta versão alia à robustez o facto de em média ser a que realiza menor número de iterações e a segunda melhor em média de cálculos da função f e média de tempo de *CPU* (a melhor das quatro versões que usam a segunda fórmula de substituição). Verifica-se ainda que obteve maior robustez quando se considerou $\rho_{\max} = 500$. Por outro lado, ao ter de calcular $\nabla\Psi_2$ analiticamente, o tempo médio por iteração é o segundo pior. De salientar ainda que em 59,3% dos 27 problemas que estamos a considerar, esta versão foi a que realizou o menor número de iterações.

Versão 5

Considerando os 27 problemas, verificamos que esta versão convergiu em todos para, pelo menos, uma combinação dos parâmetros ρ_{\max} e Δ , tendo convergido em 72,8% das 162 tentativas. Contudo, tendo uma média por problema de 67,5 iterações, 478,3 cálculos de f e 1.982 milésimos de *CPU*, verificamos que somente a versão 2 obteve piores resultados ao nível do esforço computacional. No entanto, podemos também verificar que o seu tempo médio por iteração é melhor do que os tempos das versões 3 e 4. Parece pois que caso se consiga diminuir o número médio de iterações, esta versão poderá aproximar-se dos desempenhos das versões 3 e 4.

Desta forma, pode-se concluir que a versão que conseguiu combinar a maior robustez com a melhor eficiência, terá sido a 4 - Versão Newton/Quasi-Newton(BFGS). Como segunda melhor temos a versão 3 - Versão Pseudo-Newton. No oposto, temos a versão 1 -Newton Simples- que parece bastante restritiva em relação aos problemas que é capaz de resolver. As versões 2 -Versão Quasi-Newton (BFGS)- e a versão 5-Versão Newton/Quasi-Newton (BFGS/Broyden)- tiveram desempenhos mais ou menos esperados, sendo a segunda um pouco melhor tanto ao nível da robustez como do esforço computacional.

Relativamente ao parâmetro de penalização, que depende dos valores de ρ_{\max} e Δ , verificamos que ao tentar manter o processo o mais próximo possível de uma vizinhança da região admissível ($\|c(x)\|_2 \leq 10^{-2}$), penalizando violações dessa admissibilidade ao aumentar o valor de ρ , leva a que a robustez aumente, tendo como contrapartida o aumento do esforço computacional.

7.2 Trabalho Futuro

Do trabalho realizado até ao momento e tendo em conta os resultados obtidos, é possível sugerir alguns temas de investigação futuros, designadamente:

- Testar as versões propostas em problemas de maiores dimensões.
- Construir algoritmos paralelos que possam tirar partido da estrutura do problema.
- Construir um algoritmo que permita otimizar, dentro do possível, a escolha do parâmetro de penalização.
- Testar as versões Newton/Quasi-Newton com fórmulas de substituição do vector dos multiplicadores que sejam, pelo menos, de segunda ordem.
- Deduzir fórmulas de actualização para o Jacobiano e Hessianas do vector dos multiplicadores de Lagrange ($\nabla\lambda(x)$ e $\nabla^2\lambda_i(x)$) e para as Hessianas do vector das restrições ($\nabla^2c_i(x)$) que tenham em conta a estrutura do problema, isto é, a forma do vector gradiente e da matriz Hessiana da função Ψ .
- Incorporar estas fórmulas numa implementação única e definir um método Quasi-Newton estruturado.

Bibliografia

- [Abdy-Dempster] Abdy, P.R. e Dempster, M.A.H.
Introduction to Optimization Methods
Chapman and Hall, 1974
ISBN 0-412-11040-7
- [Beltrami] Beltrami, E.J.
Technical Note - A Note on the Method of Multipliers
Journal of Optimization Theory and Applications, Vol 45, n° 3,
March 1985.
- [Bertocci-Cavalli-Spedicato] Bertocci, M; Cavalli, E. e Spedicato, E.
Computacional Performance of Diagonalized Multiplier Quasi-Newton Methods for Nonlinear Optimization With Equality Constraints
em *Numerical Optimization of Dynamic Systems*, Dixon, L. e Szegö, G. (Eds.), North-Holland Publishing Company, 1980
ISBN 0-444-85494-0
- [Byrd] Byrd, R.H.
Local Convergence of the Diagonalized Method of Multipliers
Journal of Optimization Theory and Applications, Vol 26, n°4, December 1978.
- [Celis-Dennis-Tapia] Celis, M.R.; Dennis, J.E. e Tapia, R.A.
A Trust-Region Strategy for Nonlinear Equality Constrained Optimization
em *Numerical Optimization 1984*, Boggs, P.T.; Byrd, R.H. e Schnabel (Eds.), SIAM Philadelphia (1985)
ISBN 0-89871-054-5

- [Dennis-Schnabel] Dennis, J.E. e Schnabel, R.B.
Numerical Methods for Unconstrained Optimization and Non Linear Equations
Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1983
ISBN 0-13-627216-9
- [Fernandes] Fernandes, E.M.G.P
Computação Numérica, 2ª edição
Universidade do Minho, Braga, 1998
ISBN 972-96944-1-9
- [Fletcher] Fletcher, R.
Practical Methods of Optimization, 2ª edição
John Wiley and Sons, 1991
ISBN 0-471-91547-5
- [Fontecilla] Fontecilla, R.
Inexact Secant Methods for Non Linear Constrained Optimization
SIAM Journal of Numerical Analysis, Vol. 27, pp 154-165, February 1990.
- [Glad] Glad, S.T.
Properties of Updating Methods for the Multipliers in Augmented Lagrangians
Journal of Optimization Theory and Applications, Vol 28, nº2, June 1979
- [Gill–Murray–Wright] Gill, P.E.; Murray, W. e Wright, M.
Practical Optimization
Academic Press, 1981
ISBN 0-12-283952-8 (Paperback)
- [Hock-Schittkowski] Hock, W. e Schittkowski, K.
Test Examples for Nonlinear Programming Code
Springer-Verlag, New-York, 1981

- [Lucidi] Lucidi, S.
New Results on a Class of Exact Augmented Lagrangians
Journal of Optimization Theory and Applications, Vol 58, nº2, August 1988
- [Magalhães] Magalhães, L.T.
Álgebra Linear Como Introdução à Matemática Aplicada
Texto-Editora, 1990
- [Monteiro-Fernandes] Monteiro, M.T. e Fernandes, E.M.G.P.
Método de Substituição do Vector dos Multiplicadores Baseado em Pseudo-Derivadas
Investigação Operacional, vol.17, nº1, 1997
- [Powell] Powell, M.J.D.
Methods for Nonlinear Constraints in Optimization Calculations,
em *The State of the Art in Numerical Analysis*
Iresles, A. e Powell, M.J. D. (Eds.)
Academic Press, 1986.
- [Whittler] Whittle, P.
Optimization Under Constraints - Theory and Applications of Nonlinear Programming
Wiley-Interscience
ISBN 0-471-94130-1
- [Wolfe] Wolfe, M.A.
Numerical Methods for Unconstrained Optimization
Van Nostrand Reinhold Company, 1978
ISBN 0-442-30217-7