

Optimização Não Linear

Maria Fernanda Pires da Costa

Departamento de Matemática
Escola de Ciências, Universidade do Minho

Conteúdo

1	Introdução	1
1.1	O problema de optimização não linear sem restrições	2
1.2	Notação	2
1.3	Condições de Optimalidade	3
1.3.1	Problema unidimensional ($n=1$)	5
1.3.2	Problema multidimensional ($n>1$)	5
1.4	Exercícios	7
2	Optimização não linear sem restrições	10
2.1	Forma geral do problema	10
2.2	Optimização unidimensional	12
2.2.1	Método de Procura de Davies, Swann e Campey	13
2.3	Optimização multidimensional	19
2.3.1	Métodos de procura unidimensional	20
2.3.2	CrITÉrio de paragem	28
2.3.3	Método de Newton	30
2.3.4	Método das direcções de descida máxima	36
2.3.5	Métodos quasi-Newton	38
2.4	Exercícios	46
3	Problemas de mínimos quadrados não linear	49
3.1	Classificação do problema relativamente aos resíduos	51
3.2	Métodos para problemas de mínimos quadrados não lineares	51
3.2.1	Método de Gauss-Newton	52
3.2.2	Método de Levenberg-Marquardt	54
3.2.3	Métodos quasi-Newton estruturados	55
3.2.4	Métodos quasi-Newton estruturados e factorizados	58
3.2.5	CrITÉrio de paragem	64
3.3	Exercícios	65

4	Optimização não linear com restrições	70
4.1	Forma geral do problema	70
4.2	Notação	71
4.3	Condições de Optimalidade	71
4.3.1	Condições de optimalidade de 1 ^a ordem	74
4.3.2	Condições de optimalidade de 2 ^a ordem	77
4.4	Exercícios	80
5	Programação quadrática sequencial	82
5.1	Método da programação quadrática sequencial	82
5.2	Métodos PQS práticos	86
5.3	Cálculo da direcção de procura	88
5.4	A Hessiana do modelo quadrático	89
5.4.1	Aproximações quasi-Newton	89
5.4.2	A Hessiana da Lagrangeana aumentada	91
5.5	Funções mérito e descendência	92
5.6	O efeito de Maratos	96
5.7	Exercícios	99
	Bibliografia	105

Capítulo 1

Introdução

A optimização é uma ferramenta importante na área da decisão e na análise de sistemas reais. Para a usar, primeiro tem-se que identificar alguma *função objectivo*, uma medida quantitativa do desempenho do sistema em estudo. Esta *função objectivo* pode ser lucro, tempo, energia ou qualquer quantidade ou uma combinação de quantidades que pode ser representada por um unico número. A *função objectivo* depende de certas características do sistema, designadas por variáveis ou parâmetros do sistema. O objectivo é encontrar valores para as variáveis que optimize a função objectivo, podendo algumas destas variáveis estar restringida. Matematicamente, optimização é a minimização ou maximização de uma função objectivo sujeita a restrições nas suas variáveis.

Ao processo de identificar a função objectivo, as variáveis e as restrições, de um dado problema, chama-se *modelação*. Assim, o primeiro passo no processo de optimização - e por vezes o mais importante - é a construção de um modelo matemático apropriado para o problema em estudo. Se o modelo for demasiado simplista, não se terá grande conhecimento sobre o problema real. Por outro lado, se o modelo for demasiado complexo, poderá ser difícil de o resolver.

Uma vez formulado o modelo matemático, usa-se um algoritmo de optimização para encontrar a sua solução. Em geral, o algoritmo e o modelo são complicados, sendo necessário recorrer a um computador para implementar este processo. Na prática, não existe um algoritmo universal. Em vez disso, existem vários algoritmos, sendo cada um específico para um determinado tipo de problema de optimização. É da responsabilidade do utilizador escolher qual o algoritmo mais apropriado para o problema em questão. Esta escolha é importante, e poderá determinar se o problema é resolvido rapidamente ou lentamente e se, de facto, é encontrada a solução.

Após a aplicação de um algoritmo de optimização ao modelo, tem-se que ser capaz de avaliar se o algoritmo determinou correctamente uma sua solução. Em muitos casos,

existem expressões matemáticas conhecidas como *condições de optimalidade* para verificarem que a solução encontrada, é de facto, solução do problema. Caso as condições de optimalidade não sejam verificadas, pode-se usar esta informação para melhorar a solução actual.

1.1 O problema de optimização não linear sem restrições

A formulação matemática de um problema de optimização sem restrições é

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad F(x). \quad (1.1)$$

- À função F que se pretende minimizar chama-se *função objetivo*.
- As n variáveis do problema: $x_1, x_2, \dots, x_{n-1}, x_n$ são manipuladas como um vector x , de n elementos ($x \in \mathbb{R}^n$).
- Os valores óptimos calculados pelo processo são designados por: $x^* = (x_1^*, x_2^*, \dots, x_n^*)^T$ e o correspondente valor de F é F^* .
- O problema de optimização formulado em (2.1) define um problema de *minimização*. Esta formulação não se torna restritiva uma vez que qualquer problema de *maximização* pode ser colocado nesta forma. Ou seja, pode ser resolvido implementando um método que calcule mínimos de funções, pois

$$\text{máximo } F(x) = - \text{mínimo } [-F(x)]$$

e o valor de x , onde F atinge o seu máximo é o mesmo onde $-F$ atinge o seu mínimo.

1.2 Notação

As derivadas de primeira e segunda ordens de F definem os seguintes vector e matriz.

A derivada parcial de F em ordem a x_i ($i = 1, 2, \dots, n$) é

$$\frac{\partial F}{\partial x_i}$$

e é uma função de x .

Chama-se vector gradiente de F , e será denotado por ∇F , ao vector formado pelas n derivadas parciais de 1^{as} ordem de F :

$$\nabla F(x) = \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \dots \\ \frac{\partial F}{\partial x_n} \end{pmatrix}. \quad (1.2)$$

O vector gradiente é normal ao hiperplano tangente à curva F , no ponto x .
Este hiperplano pode ser definido pelo vector gradiente e pelo escalar $F(x) \in R$.

As segundas derivadas parciais de F representam-se por

$$\frac{\partial^2 F}{\partial x_i \partial x_j} \text{ para } i \neq j \quad \text{e} \quad \frac{\partial^2 F}{\partial x_i^2} \text{ para } i = j$$

e são funções de x .

Chama-se matriz Hessiana de F , e será denotada por $\nabla^2 F$, à matriz quadrada e simétrica de ordem n formada pelas 2^{as} derivadas parciais de F :

$$\nabla^2 F(x) = \begin{pmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \dots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_n^2} \end{pmatrix}. \quad (1.3)$$

1.3 Condições de Optimalidade

Dado que o conhecimento sobre a função F , em muitos dos métodos de optimização, é geralmente apenas local, a optimalidade de um ponto x^* é definida pela sua relação com os pontos da vizinhança. Define-se uma vizinhança de x^* como sendo um conjunto aberto que contém x^* e que está contida no domínio de F .

Para problemas sem restrições, o domínio de F define o conjunto de pontos admissíveis. Vai denotar-se por $\mathcal{D} \subseteq \mathbb{R}^n$ a região admissível de F .

Definição 1.1 Um ponto x^* diz-se minimizante local fraco do problema (2.1) se existe uma vizinhança \mathbb{V} de x^* tal que $F(x^*) \leq F(x)$ para todo $x \in \mathbb{V}$.

Definição 1.2 Um ponto x^* diz-se minimizante local forte (também designado por minimizante local estrito) do problema (2.1) se existe uma vizinhança \mathbb{V} de x^* tal que $F(x^*) < F(x)$ para todo $x \in \mathbb{V}$ com $x \neq x^*$.

Definição 1.3 Um ponto x^* diz-se minimizante global do problema (2.1) se $F(x^*)$ atinge o seu valor mais baixo no domínio de F .

Quando a função objectivo F é uma função convexa e a região admissível Ω é um conjunto convexo, uma solução local é também global.

Definição 1.4 Um conjunto \mathcal{D} diz-se convexo se, para quaisquer elementos x e y de \mathcal{D} ,

$$ax + (1 - a)y \in \mathcal{D}, \quad \text{para todo } 0 \leq a \leq 1.$$

Por outras palavras, se x e y estão em \mathcal{D} , então o segmento de recta que liga x e y está também em \mathcal{D} .

Definição 1.5 Uma função F diz-se convexa num conjunto convexo \mathcal{D} se satisfaz

$$F(ax + (1 - a)y) \leq aF(x) + (1 - a)F(y) \quad (1.4)$$

para todo $0 \leq a \leq 1$ e para todo $x, y \in \mathcal{D}$.

Esta definição diz que o segmento de recta que liga os pontos $(x, F(x))$ e $(y, F(y))$ permanece sobre ou acima do gráfico da função. A função F diz-se estritamente convexa se a desigualdade em (1.4) for estrita. De modo análogo, se define uma função côncava.

Definição 1.6 Uma função F diz-se côncava num conjunto convexo \mathcal{D} se satisfaz

$$F(ax + (1 - a)y) \geq aF(x) + (1 - a)F(y)$$

para todo $0 \leq a \leq 1$ e para todo $x, y \in \mathcal{D}$.

Quando a função F é continuamente diferenciável, existem maneiras mais eficientes e práticas para a identificação de um minimizante local. Em particular, se F é duas vezes continuamente diferenciável, é possível saber se x^* é um minimizante local analisando as derivadas de primeira e segunda ordens da função F em x^* , isto é, analisando o gradiente ∇F e a Hessiana $\nabla^2 F$ em x^* .

1.3.1 Problema unidimensional (n=1)

Se a função objectivo é continuamente diferenciável até à segunda ordem e um mínimo local de F existe num ponto finito x^* , então as seguintes condições devem ser verificadas.

- *Condições necessárias para a existência de um mínimo num problema unidimensional sem restrições:*

- $F'(x^*) = 0$
- $F''(x^*) \geq 0$.

A primeira condição diz-se de *primeira ordem* e define os *pontos estacionários* da função F . A primeira derivada de F também se anula em pontos que são máximos locais e pontos de inflexão.

A segunda condição, porque envolve a segunda derivada de F , é conhecida por *condição de segunda ordem*.

Para garantir que um certo ponto é óptimo, devem verificar-se as condições suficientes.

- *Condições suficientes para a existência de um mínimo num problema unidimensional sem restrições:*

- $F'(x^*) = 0$
- $F''(x^*) > 0$.

Se F é contínua, mas x^* é um ponto de descontinuidade de F' , as condições suficientes para que x^* seja um mínimo local forte são:

$$F'(x^* + \varepsilon) > 0 \quad \text{e} \quad F'(x^* - \varepsilon) < 0$$

em que ε é uma quantidade positiva e próxima de zero.

1.3.2 Problema multidimensional (n>1)

As condições necessárias de optimalidade são deduzidas assumindo que x^* é minimizante local e a seguir deduzem-se propriedades sobre $\nabla F(x^*)$ e $\nabla^2 F(x^*)$.

Teorema 1.1 (*Condição necessária de primeira ordem*)

Se x^ é um minimizante local e F é continuamente diferenciável numa vizinhança aberta de x^* , então $\nabla F(x^*) = 0$.*

Prova. Veja-se no Teorema 2.2 em [8]. ■

Teorema 1.2 (*Condição necessária de segunda ordem*)

Se x^* é um minimizante local e $\nabla^2 F$ é contínua numa vizinhança aberta de x^* , então $\nabla F(x^*) = 0$ e $\nabla^2 F(x^*)$ é semidefinida positiva¹.

Prova. Veja-se no Teorema 2.3 em [8]. ■

Qualquer ponto que satisfaz $\nabla F(x) = 0$ designa-se por ponto estacionário de F . Note-se que ∇F pode anular-se num maximizante local de F , ou então num ponto que não é minimizante nem maximizante, conhecido por ponto sela ou de descanso.

As condições necessárias são, em geral, usadas para mostrar que um dado ponto não é ótimo.

As condições suficientes, que a seguir se enunciam, se verificadas no ponto x^* , garantem que x^* é um minimizante local.

Teorema 1.3 (*Condições suficientes de segunda ordem*)

Se $\nabla^2 F$ é contínua numa vizinhança aberta de x^* e se $\nabla F(x^*) = 0$ e $\nabla^2 F(x^*)$ é definida positiva², então x^* é um minimizante local forte de F .

Prova. Veja-se no Teorema 2.4 em [8]. ■

Se a função F^3 é convexa, os minimizantes locais e globais são simples de caracterizar.

Teorema 1.4 *Quando F é convexa, qualquer minimizante local x^* é um minimizante global de F . Além disso, se F é diferenciável, então qualquer ponto estacionário é um minimizante global de F .*

¹Se os determinantes das submatrizes principais de uma matriz são não negativos, a matriz diz-se semidefinida positiva.

²Se os determinantes das submatrizes principais de uma matriz são positivos, a matriz diz-se definida positiva.

³**Teorema:** Seja F uma função duas vezes diferenciável. F é convexa $\Leftrightarrow \forall x \in DF, \nabla^2 F(x)$ é semidefinida positiva.

Prova. Veja-se no Teorema 2.5 em [8]. ■

Nota: As condições necessária e suficiente de segunda ordem para a existência de um máximo no ponto x^* são, respectivamente, $\nabla^2 F(x^*)$ semidefinida negativa e definida negativa⁴. Uma matriz que não é definida positiva, semidefinida positiva, definida negativa nem semidefinida negativa, diz *indefinida*. Se a Hessiana é indefinida, num ponto estacionário, então esse ponto é ponto sela.

Estes resultados fornecem as bases para os algoritmos de otimização sem restrições. De uma maneira ou de outra, todos os algoritmos procuram um ponto para o qual ∇F se anula.

1.4 Exercícios

1. Considere a função $F : \mathbb{R} \rightarrow \mathbb{R}$ definida por

$$F(x) = 15 - 12x - 25x^2 + 2x^3.$$

- (a) Use as primeira e segunda derivadas para determinar o máximo local e mínimo local de F .
- (b) Mostre que F não tem nem um máximo global nem um mínimo global.

2. Considere a função $F : \mathbb{R} \rightarrow \mathbb{R}$ definida por

$$F(x) = 3x^3 + 7x^2 - 15x - 3.$$

Calcule todos os pontos estacionários desta função, e determine se são mínimos locais e máximos locais. Esta função tem um máximo global ou um mínimo global?

3. Considere a função $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ definida por

$$F(x_1, x_2) = \frac{1}{3}x_1^3 + \frac{1}{2}x_1^2 + 2x_1x_2 + \frac{1}{2}x_2^2 - x_2 + 9.$$

Calcule todos os pontos estacionários desta função, e verifique se são mínimos locais e máximos locais. Esta função tem um máximo global ou um mínimo global?

⁴Se os determinantes das submatrizes principais de uma matriz têm sinais alternadamente negativos e positivos, sendo o primeiro negativo, a matriz diz-se definida negativa. Se alguns destes determinantes forem nulos, a matriz é semidefinida negativa.

4. Considere a função $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ definida por

$$F(x_1, x_2) = 8x_1^2 + 3x_1x_2 + 7x_2^2 - 25x_1 + 31x_2 - 29.$$

Calcule todos os pontos estacionários desta função, e verifique se são mínimos locais e máximos locais. Esta função tem um máximo global ou um mínimo global?

5. Mostre que qualquer ponto da linha $x_2 - 2x_1 = 0$ é um mínimo de $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ definida por

$$F(x_1, x_2) = 4x_1^2 - 4x_1x_2 + x_2^2.$$

6. Verifique se o ponto $(0, -1)^T$ é mínimo da função

$$F(x_1, x_2) = x_1 + x_2 + \frac{1}{2}(x_1^2 + x_2^2) - \frac{x_1 + x_1^2 + x_1^3}{1 + x_1^4}.$$

Justifique.

7. Considere o problema:

$$\text{minimizar } F(x_1, x_2) = (x_2 - x_1^2)(x_2 - 2x_1^2).$$

- (a) Mostre que as condições necessárias de optimalidade são satisfeitas no ponto $(0, 0)^T$.
- (b) Mostre que a origem é um mínimo local de F para qualquer linha que passe na origem (que é, $x_2 = mx_1$).
- (c) Mostre que a origem não é um mínimo local de F (considere, por exemplo, curvas da forma $x_2 = kx_1^2$). Que conclusões se pode retirar a partir isto?

8. Considere o problema

$$\text{minimizar } F(x_1, x_2) = (x_1 - 2x_2)^2 + x_1^4.$$

Determine o mínimo de F . Verifique que a condição necessária de 2ª ordem para um mínimo local é satisfeita nesse ponto. Verifica a condição suficiente de 2ª ordem? Este ponto é um minimizante local estrito? É um mínimo global?

9. Seja

$$F(x_1, x_2) = 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4.$$

Determine os minimizantes/maximizantes de F e indique que tipo de mínimos ou máximos são (local, global, estrito, etc).

10. Seja

$$F(x_1, x_2) = cx_1^2 + x_2^2 - 2x_1x_2 - 2x_2$$

onde c é um escalar.

- (a) Determine os pontos estacionários de F para cada valor de c .
- (b) Para que valores de c pode F ter um minimizante? Para que valores de c pode F ter um maximizante? Determine os minimizantes/maximizantes correspondentes a tais valores de c , e indique que tipo de mínimos ou máximos são (local, global, estrito, etc).

Capítulo 2

Optimização não linear sem restrições

2.1 Forma geral do problema

A formulação matemática de um problema de optimização, na sua forma mais geral, é

$$\text{minimizar}_{x \in R^n} F(x) \tag{2.1}$$

$$\begin{aligned} \text{sujeito a } h_i(x) &= 0, & i = 1, \dots, m \\ h_j(x) &\geq 0, & j = 1, \dots, \overline{m} \end{aligned}$$

onde F é a função objectivo, as equações $h_i(x) = 0$, $i = 1, \dots, m$ são as restrições nas variáveis do tipo igualdade e as inequações $h_j(x) \geq 0$, $j = 1, \dots, \overline{m}$ são as restrições nas variáveis do tipo de desigualdade. Qualquer conjunto de equações e inequações pode ser escrito na forma (2.1). Em particular, qualquer inequação do tipo “menor ou igual” pode ser transformada numa restrição equivalente do tipo “maior ou igual”. Tais transformações são meramente “cosméticas”, mas simplificam a notação para descrever as restrições.

Podem existir problemas de optimização onde só existam restrições de igualdade, outros onde só existam restrições de desigualdades e ainda outros sem qualquer tipo de restrição nas variáveis.

O problema de otimização (2.1) define um problema de *minimização*. Esta formulação não se torna restritiva uma vez que qualquer problema de *maximização* pode ser formulado como sendo um problema de minimização, pois

$$\begin{aligned} & \text{maximizar}_{x \in R^n} G(x) \\ & \text{sujeito a } \begin{aligned} h_i(x) &= 0, & i = 1, \dots, m \\ \tilde{h}_j(x) &\leq 0, & j = 1, \dots, \overline{m} \end{aligned} \end{aligned}$$

é equivalente a

$$\begin{aligned} & \text{minimizar}_{x \in R^n} F(x) \equiv -G(x) \\ & \text{sujeito a } \begin{aligned} h_i(x) &= 0, & i = 1, \dots, m \\ h_j(x) &\equiv -\tilde{h}_j(x) \geq 0, & j = 1, \dots, \overline{m}. \end{aligned} \end{aligned}$$

Um ponto x que verifique todas as restrições, chama-se um ponto um *ponto admissível*. Ao conjunto de todos admissíveis chama-se *conjunto admissível* ou *região admissível*. Assim,

$$\mathcal{D} = \{x \in R^n : h_i(x) = 0, \quad i = 1, \dots, m \text{ e } h_j(x) \geq 0, \quad j = 1, \dots, \overline{m}\}$$

Classificação dos problemas (mais usuais):

- Problemas unidimensionais ($n = 1$, ou seja, $x \in R$)
- Problemas multidimensionais ($n > 1$, ou seja, $x = (x_1, x_2, \dots, x_n)^T \in R^n$)
- Problemas Sem Restrições (ou seja, $\text{minimizar}_{x \in R^n} F(x)$)
- Problemas Com Restrições (por exemplo, a formulação genérica dada por (2.1)):

– Problemas de Programação Linear

(F e h são funções lineares, isto é, $F(x) = d^T x$, $h(x) = Ax - b$, onde A é uma matriz constante do tipo $(m + \overline{m}) \times n$, $d \in R^n$ e $b \in R^{m+\overline{m}}$ são vectores constantes. A linha t de A contém os coeficientes da restrição t , para $t = 1, \dots, m + \overline{m}$.)

Problemas de Programação Quadrática

(F é uma função quadrática, isto é, $F(x) = \frac{1}{2}x^T Gx + d^T x$ e h são funções lineares, onde G é uma matriz simétrica constante do tipo $n \times n$ e d é um vector constante).

Nota: o valor óptimo dos problemas de PL e PQ não são afectados por qualquer termo constante na função objectivo, por isso ele é ignorado na formulação destes problemas.

Problemas com Limites Simples

(restrições nas variáveis do tipo $l_i \leq x_i \leq u_i$, $i = 1, 2, \dots, n$, e qualquer dos limites pode não estar presente em algumas das variáveis: $-\infty \leq l_i$, $u_i \leq \infty$.)

Problemas de Programação Não Linear

(pelo menos uma das funções envolvidas, $F(x)$, $h(x)$ é não linear.)

2.2 Optimização unidimensional

A formulação matemática de um problema de optimização unidimensional, sem restrições é

$$\text{minimizar}_{x \in R} F(x).$$

Os métodos para resolução de problemas de minimização unidimensionais podem ser agrupados em duas classes:

Métodos de aproximação - normalmente aplicáveis a funções contínuas, aproximam a função F por uma outra mais simples, que será posteriormente analisada para se determinar o seu mínimo. As funções de aproximação são, em geral, funções polinomiais de grau baixo.

Métodos de procura directa - partem de um intervalo que contém o mínimo e têm como objectivo reduzir esse intervalo, calculando e comparando valores de F em diversos pontos desse intervalo. Estes métodos podem ser aplicados a qualquer função, desde que ela seja unimodal.

Definição 2.1 Uma função $F \in R$ diz-se unimodal quando x^* é o único valor da variável para o qual $F(x^*) < F(x)$, para todo o x do intervalo que contém x^* .

Nesta secção apenas apresenta-se um método da classe dos métodos de procura directa, nomeadamente o método de procura de Davies, Swann e Campey. Este método em si, engloba os métodos de aproximação baseados na interpolação quadrática e cúbica.

2.2.1 Método de Procura de Davies, Swann e Campey

Primeiro, constrói, em cada iteração, 3 ou 4 pontos igualmente espaçados que definem um intervalo que contém o mínimo.

A seguir, aproxima a função F nesse intervalo, por uma quadrática ou cúbica, com o objectivo de calcular o seu mínimo. O mínimo da quadrática ou cúbica é uma aproximação ao mínimo de F .

Procura do intervalo que contém o mínimo:

A procura começa a partir de uma aproximação ao mínimo de F , x_1 , e um incremento Δ . A partir do x_1 e no sentido positivo, vai-se construindo uma sequência de pontos: $x_2, x_3, x_4, x_5, \dots$ distanciados uns dos outros de, respectivamente: $\Delta, 2\Delta, 4\Delta, 8\Delta, \dots$, até ser encontrado um ponto, para o qual o valor da função F tenha aumentado quando comparado com o valor da função no ponto anterior.

x_1	$F(x_1)$	
$x_2 = x_1 + p\Delta$	$F(x_2)$	$F(x_2) \leq F(x_1)$
$x_3 = x_2 + p2\Delta$	$F(x_3)$	$F(x_3) \leq F(x_2)$
$x_4 = x_3 + p4\Delta$	$F(x_4)$	$F(x_4) \leq F(x_3)$
\dots	\dots	\dots
$x_i = x_{i-1} + p2^{i-2}\Delta$	$F(x_i)$	$F(x_i) > F(x_{i-1})$

com $p = 1$ quando a procura corre no sentido positivo. Se a procura corre no sentido negativo $p = -1$. Quando a procura pará no ponto x_i tem-se

$$x_{i-2} < x_{i-1} < x_i, \quad \text{quando } p = 1$$

ou

$$x_i < x_{i-1} < x_{i-2}, \quad \text{quando } p = -1$$

em que

$$F(x_{i-2}) \geq F(x_{i-1}) \text{ e } F(x_{i-1}) < F(x_i).$$

Como a distância entre x_i e x_{i-1} é duas vezes a distância entre x_{i-1} e x_{i-2} , o último ponto

a calcular é o ponto médio do último intervalo:

$$x_m = \frac{x_i + x_{i-1}}{2}.$$

Portanto, tem-se 4 pontos igualmente espaçados

$$x_{i-2} < x_{i-1} < x_m < x_i, \quad \text{quando } p = 1 \quad (2.2)$$

ou

$$x_i < x_m < x_{i-1} < x_{i-2}, \quad \text{quando } p = -1. \quad (2.3)$$

Após serem conhecidos os pontos de (2.2) ou (2.3), a etapa seguinte consiste em aproximar a função F , por uma quadrática ou por uma cúbica, no intervalo definido pelos pontos indicados em (2.2) ou (2.3).

Nota: Se, a partir de x_1 , $F(x_2 = x_1 + \Delta) > F(x_1)$ a procura deve voltar-se para o sentido negativo, a começar novamente por x_1 . O próximo ponto, na procura, é $\bar{x}_2 = x_1 - \Delta$. Se $F(\bar{x}_2) > F(x_1)$, isso significa que o intervalo definido por $[\bar{x}_2, x_2]$, com x_1 como ponto médio, contém o mínimo desejado. Neste caso, aproxima-se a função, nesse intervalo, por um polinómio de grau 2.

Interpolação cúbica:

Com os quatro pontos referidos em (2.2) ou em (2.3) e os correspondentes valores de F , determina-se o polinómio de grau 3 que passa por esses pontos:

$$p_3(x) = c_0 + c_1x + c_2x^2 + c_3x^3.$$

Os coeficientes do polinómio calculam-se a partir da resolução do sistema linear:

$$\begin{cases} c_0 + c_1x_{i-2} + c_2x_{i-2}^2 + c_3x_{i-2}^3 = F(x_{i-2}) \\ c_0 + c_1x_{i-1} + c_2x_{i-1}^2 + c_3x_{i-1}^3 = F(x_{i-1}) \\ c_0 + c_1x_m + c_2x_m^2 + c_3x_m^3 = F(x_m) \\ c_0 + c_1x_i + c_2x_i^2 + c_3x_i^3 = F(x_i) \end{cases}$$

Os pontos estacionários de $p_3(x)$ calculam-se a partir de:

$$\frac{dp_3(x)}{dx} = c_1 + 2c_2x + 3c_3x^2 = 0$$

donde se retira que

$$x^*(c) = \frac{-c_2 \pm \sqrt{c_2^2 - 3c_1c_3}}{3c_3}.$$

Portanto, tem-se dois pontos estacionários. O mínimo da cúbica, $x^*(c)$, será o que torna a segunda derivada $\frac{d^2 p_3(x)}{d\alpha^2} = 2c_2 + 6c_3x$ positiva, e será então usado para aproximar o mínimo da função F .

Interpolação quadrática:

Para a aproximação quadrática, torna-se necessário seleccionar 3 dos 4 pontos. Esta selecção é baseada na comparação entre os valores de F , nos dois pontos interiores do intervalo. Vamos assumir que os pontos escolhidos são os pontos (2.2):

$F(x_{i-1}) \leq F(x_m) \implies$	pontos escolhidos: x_{i-2}, x_{i-1}, x_m (fazer: $x_1 = x_{i-2}, x_2 = x_{i-1}$ e $x_3 = x_m$)
senão \implies	pontos escolhidos: x_{i-1}, x_m, x_i . (fazer: $x_1 = x_{i-1}, x_2 = x_m$ e $x_3 = x_i$)

Com qualquer destes conjuntos de 3 pontos e os correspondentes valores da função F , determina-se o polinómio de grau 2 que passa por eles.

$$p_2(x) = c_0 + c_1x + c_2x^2.$$

Os coeficientes do polinómio calculam-se a partir da resolução do sistema linear:

$$\begin{cases} c_0 + c_1x_1 + c_2x_1^2 + c_3x_1^3 = F(x_1) \\ c_0 + c_1x_2 + c_2x_2^2 + c_3x_2^3 = F(x_2) \\ c_0 + c_1x_3 + c_2x_3^2 + c_3x_3^3 = F(x_3) \end{cases} \quad (2.4)$$

O mínimo desta quadrática satisfaz:

$$\frac{dp_2(x)}{dx} = c_1 + 2c_2x = 0$$

donde se retira que

$$x^*(q) = \frac{-c_1}{2c_2}. \quad (2.5)$$

Como os 3 pontos encontram-se igualmente distanciados de uma quantidade igual a $\delta \equiv 2^{i-3}\Delta$ e como $x_1 < x_2 < x_3$, da resolução do sistema (2.4) e (2.5) tem-se:

$$x^*(q) = x_2 + \delta \frac{F(x_1) - F(x_3)}{2[F(x_3) - 2F(x_2) + F(x_1)]}.$$

Como o mínimo, da quadrática ou da cúbica, não coincide com o mínimo da função F , o processo terá de ser repetido, originando um processo iterativo. A nova iteração inicia-se

pela fase de procura de um intervalo que contenha o mínimo e termina com a aproximação. O recomeço tem como ponto de partida o mínimo, $x^*(q)$ ou $x^*(c)$, da iteração anterior, e deve usar um incremento, Δ , menor do que o usado na iteração anterior. A constante de decréscimo de Δ é $M < 1$. Este processo iterativo, correspondente à implementação do método de procura de Davies, Swann e Campey, termina quando o intervalo entre os pontos Δ , para a aproximação, for inferior a uma precisão predeterminada ε .

Exemplo 2.1 Dada a função $F : R \rightarrow R$ definida por

$$F(x) = -500x + \pi x^3$$

calcule o seu mínimo usando o método de procura DSC baseado na interpolação quadrática. O processo deve ser iniciado com $x_1 = 7.5$. Considere $\Delta = 0.1$, $M = \frac{1}{2}$ e $\varepsilon = 0.1$.

- 1ª iteração:

Procura do intervalo

$$\begin{array}{ll} x_1 = 7.5 & F_1 = -2.4246e + 003 \\ x_2 = x_1 + \Delta = 7.6 & F_2 = -2.4209e + 003 \quad F_2 > F_1 \end{array}$$

Como $F_2 > F_1$, inverte-se a procura:

$$\begin{array}{ll} \bar{x}_2 = x_1 - \Delta = 7.4 & \bar{F}_2 = -2.4270e + 003 \quad \bar{F}_2 \leq F_1 \\ \bar{x}_3 = \bar{x}_2 - 2\Delta = 7.2 & \bar{F}_3 = -2.4274e + 003 \quad \bar{F}_3 \leq \bar{F}_2 \\ \bar{x}_4 = \bar{x}_2 - 4\Delta = 6.8 & \bar{F}_4 = -2.4122e + 003 \quad \bar{F}_4 > \bar{F}_3 \end{array}$$

Como $\bar{F}_4 > \bar{F}_3$, para-se a procura:

$$\begin{array}{ll} x_m = \frac{\bar{x}_4 + \bar{x}_3}{2} = 7.0 & F_m = -2.4224e + 003 \end{array}$$

Interpolação quadrática

Os pontos escolhidos para a interpolação são: $x_1 = 7.0$, $x_2 = 7.2$ e $x_3 = 7.4$ com $F_1 = -2.4224e + 003$, $F_2 = -2.4274e + 003$ e $F_3 = -2.4270e + 003$. O espaçamento entre os pontos é $\delta = 0.2$. Vem então que

$$x^*(q) = 7.2 + 0.2 \frac{-2.42224e + 003 + 2.4270e + 003}{2[-2.4270e + 003 - 2(-2.4274e + 003) - 2.4224e + 003]} = 7.2852$$

valor que deve iniciar a próxima iteração, uma vez que $\Delta = 0.1 < \varepsilon = 0.1$. Na próxima iteração tem-se então $x_1 = 7.2852$ e $\Delta = 0.1 \times 0.5 = 0.05$.

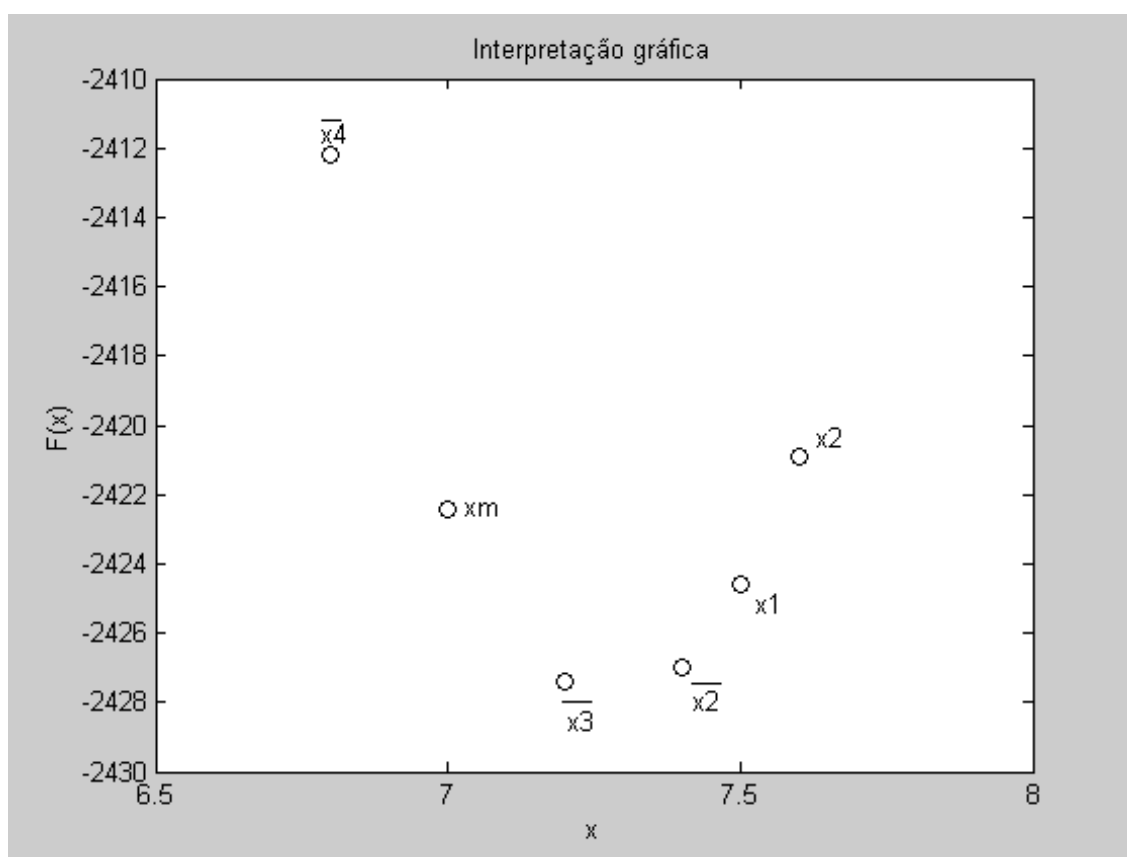


Figura 2.1: Pontos obtidos na 1ª iteração

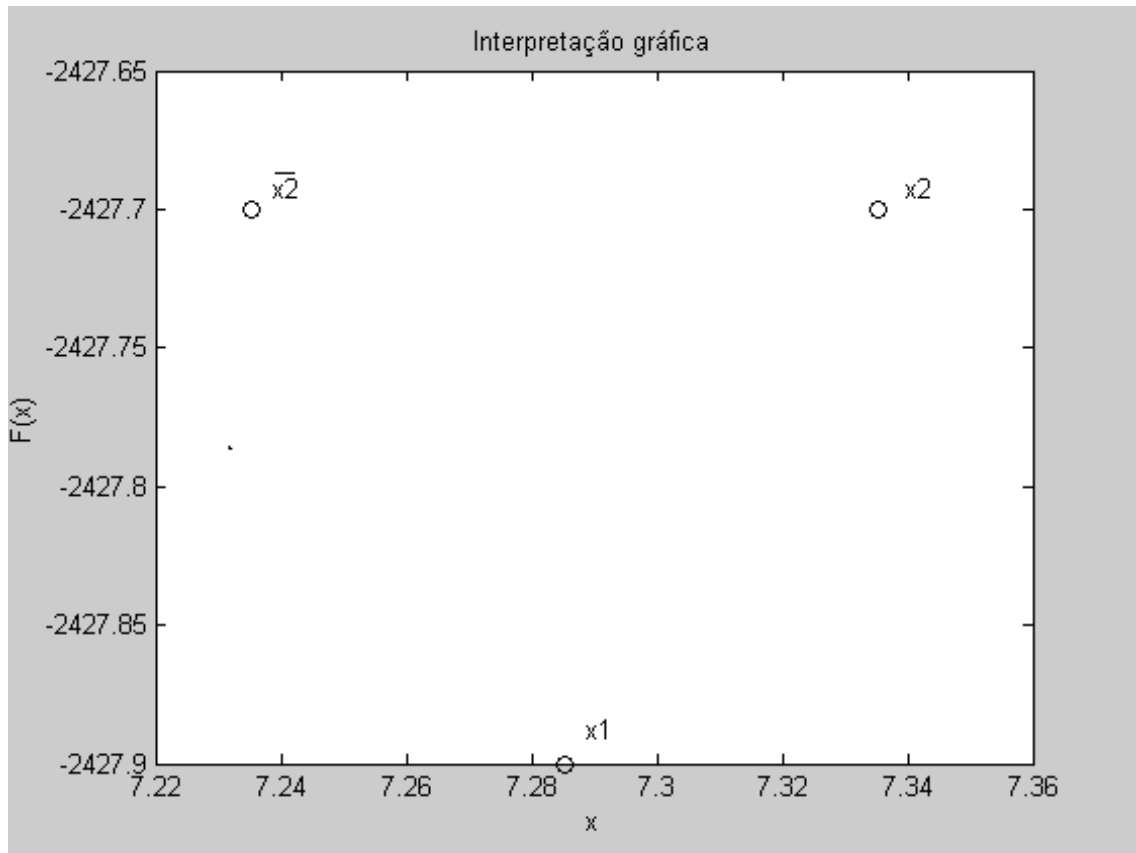


Figura 2.2: Pontos obtidos na 2ª iteração

- 2ª iteração:

Procura do intervalo

$$x_1 = 7.2852$$

$$F_1 = -2.4279e + 003$$

$$x_2 = x_1 + \Delta = 7.3352$$

$$F_2 = -2.4277e + 003 \quad F_2 > F_1$$

Como $F_2 > F_1$, inverte-se a procura:

$$\bar{x}_2 = x_1 - \Delta = 7.2352$$

$$\bar{F}_2 = -2.4277e + 003 \quad \bar{F}_2 > F_1$$

Como $\bar{F}_2 > F_1$, pará-se a procura.

Interpolação quadrática

Tem-se três pontos igualmente espaçados e procede-se à interpolação: $x_1 = 7.2352$, $x_2 = 7.2852$ e $x_3 = 7.3352$ com $F_1 = -2.4270e + 003$, $F_2 = -2.4279e + 003$ e

$F_3 = -2.4277e + 003$. O espaçamento entre os pontos é $\delta = 0.05$. Vem então que

$$x^*(q) = 7.2852 + 0.05 \frac{-2.4277e + 003 + 2.4277e + 003}{\dots} = 7.2852$$

Como $\Delta = 0.05 < \epsilon = 0.1$ pára-se com $x^* = 7.2852$. É de referir que, a solução analítica deste problema é $\sqrt{\frac{500}{3\pi}} = 7.283656$.

2.3 Optimização multidimensional

A formulação matemática de um problema multidimensional sem restrições é dada por

$$\text{minimizar}_{x \in R^n} F(x).$$

Neste problema vamos assumir que a função F é continuamente diferenciável até à 2ª ordem.

Os métodos para a resolução de problemas de optimização não linear sem restrições, podem ser agrupados em duas classes:

- **Métodos de procura** - usam apenas valores da função.

Métodos do gradiente – usam valores da função e também informação relativa às derivadas de F :

- ☐ vector gradiente e/ou
- ☐ matriz Hessiana.

Se apenas usa o vector gradiente o método é caracterizado como sendo método de 1ª ordem; se também usa a matriz Hessiana o método é designado por método de 2ª ordem.

Os métodos do gradiente baseiam-se na expansão em série de Taylor de F , no ponto $x + d$,

$$F(x + d) = F(x) + \nabla F(x)^T d + \frac{1}{2} d^T \nabla^2 F(x) d + \dots$$

Neste capítulo vamos apresentar alguns métodos da classe dos métodos do gradiente.

Nota: As funções não diferenciáveis são, em geral, mais difíceis de minimizar do que as funções continuamente diferenciáveis. Existem métodos específicos, pertencentes à

classe dos métodos de procura, para minimizar problemas com descontinuidades na função objectivo e outros para minimizar problemas com descontinuidades bem estruturadas no gradiente. Se a função objectivo tem poucas descontinuidades no gradiente e se elas não ocorrem na vizinhança do mínimo, então os métodos normalmente usados para problemas continuamente diferenciáveis são os mais eficientes.

2.3.1 Métodos de procura unidimensional

Os métodos do gradiente são métodos iterativos, e são caracterizados por uma equação iterativa que, a partir de uma aproximação inicial x_1 , ao mínimo x^* da função F , constrói uma sucessão $\{x_k\}$, $k = 1, 2, \dots$ de aproximações a x^* .

Impostas certas condições em x_1 e F , a sucessão de aproximações converge para x^* .

Definição 2.2 *Seja $F : R^n \rightarrow R$ uma função com primeiras derivadas parciais contínuas num ponto $\bar{x} \in R^n$ e seja $d \in R^n$ um vector dado não nulo. Seja $\nabla F(\bar{x})$ o vector gradiente de F , calculado no ponto \bar{x} .*

O vector d define-se como sendo uma direcção de descida, em relação a F , em \bar{x} , se

$$\nabla F(\bar{x})^T d < 0.$$

Se a direcção d for de descida, o valor da função diminui à medida que nos afastamos de \bar{x} , ao longo dessa direcção. Isto é, existe um escalar α positivo, tal que

$$F(\bar{x} + \alpha d) < F(\bar{x}).$$

O escalar α dá-nos o comprimento do deslocamento que devemos efectuar ao longo da direcção, d .

Ao procedimento que calcula o valor do escalar α chama-se *procura unidimensional*.

Estas ideias sugerem uma classe de métodos para minimizar F . Estes métodos, conhecidos por **métodos das direcções de descida** podem ser descritos, na generalidade, pelos seguintes passos:

Algoritmo:

1. Dar x_1 e calcular $F(x_1)$ e $\nabla F(x_1)$. Fazer $k = 1$.
2. Calcular o vector direcção d_k de tal forma que $\nabla F(x_k)^T d_k < 0$.
3. Calcular o escalar α_k de tal modo que $F(x_k + \alpha_k d_k) < F(x_k)$.

4. Calcular $x_{k+1} = x_k + \alpha_k d_k$, $F(x_{k+1})$ e $\nabla F(x_{k+1})$.
5. Se x_{k+1} satisfaz o critério de paragem, terminar.
6. (senão) Fazer $k = k + 1$ e voltar para 2.

Existe uma grande variedade de métodos de optimização para calcular o vector direcção d_k e o escalar α_k que satisfazem as condições dos passos 2. e 3. do algoritmo.

Procura unidimensional exacta

Calcula um valor de α que minimiza, localmente, a função F ao longo da direcção d_k . Estes cálculos originam um *valor óptimo* do comprimento do deslocamento a efectuar ao longo de d_k .

A condição de minimizar $F(x_k + \alpha d_k)$, em relação a α , define um problema de minimização unidimensional, e é muito mais restrita do que a condição imposta pelo passo 3 do algoritmo.

Para determinar α_k , na iteração k , que minimiza F como uma função apenas de α

$$\underset{\alpha \in \mathbb{R}}{\text{minimizar}} \quad \phi(\alpha) \equiv F(x_k + \alpha d_k), \quad (2.6)$$

de acordo com a condição necessária de 1ª ordem, resolve-se:

$$\phi'(\alpha) = 0.$$

Embora um valor de α_k que satisfaz: $F(x_k + \alpha_k d_k) = \min_{\alpha \in \mathbb{R}} F(x_k + \alpha d_k)$ produz uma diminuição óptima no valor da função F , em cada iteração do algoritmo com direcções de descida, a procura unidimensional efectivamente exacta é na maior parte dos casos impossível.

- Só em casos muito particulares é que a equação $\phi'(\alpha) = 0$ pode ser resolvida exactamente.

Nos outros casos, usam-se algoritmos de aproximação polinomial: quadrática ou cúbica, para determinar uma aproximação ao valor óptimo de α , isto é:

- aproxima-se a função ϕ por um polinómio em α de grau 2 ou 3;
- a seguir, determina-se analiticamente o mínimo dessa aproximação polinomial.

O método de Davies, Swann e Campey (DSC) é um dos métodos mais conhecidos e usados para estimar o α_k .

Desvantagem: A fase de procura de um intervalo que contenha o mínimo pode obrigar a muitos cálculos de ϕ , tornando-o pouco eficiente.

Procura unidimensional não exacta

A maior parte do trabalho de computação necessário para implementar muitos dos algoritmos com direcções de descida reside na determinação do α_k .

Dado que a eficiência computacional da procura unidimensional é determinante para o desempenho do algoritmo de descida, têm sido dedicados esforços no desenvolvimento de técnicas eficazes e rápidas de procura unidimensional.

Por vezes, métodos mais simples e menos precisos são preferidos, uma vez que uma redução considerada significativa, no valor da função se torna suficiente para garantir a convergência do algoritmo.

Assim, na determinação do comprimento do passo α_k , tem-se dois objectivos: Escolher α_k de modo a obter-se uma redução significativa no valor de F , mas ao mesmo tempo, não perder muito tempo a fazer esta escolha.

Os procedimentos de procura unidimensional típicos, experimentam uma sucessão de valores para α , e param e aceitam um desses elementos da sucessão, quando certas condições são satisfeitas. A procura unidimensional é feita em duas fases, nomeadamente: Uma fase de procura de um intervalo que contenha os comprimentos do passo aceitáveis, e uma fase de bissecção que calcula, dentro deste intervalo, o melhor comprimento do passo.

A seguir, apresentam-se as várias condições de terminação para o procedimento de procura unidimensional e mostra-se que os comprimentos do passo não necessitam de permanecer perto dos mínimos da função unidimensional $\phi(\alpha)$ definida em (2.6).

Condições de Wolfe Uma condição de procura unidimensional não exacta mais conhecida estipula que α_k deve, em primeiro lugar, originar uma *redução significativa* no valor da função objectivo F , dada pela seguinte desigualdade:

$$\underbrace{F(x_k + \alpha d_k)}_{\phi(\alpha)} \leq \underbrace{F(x_k) + \mu_1 \alpha \nabla F(x_k)^T d_k}_{l(\alpha)} \quad (2.7)$$

onde a constante $\mu_1 \in (0, 1)$. Esta condição é conhecida por *condição de Armijo*. A condição de redução significativa está ilustrada na figura 2.3.

O lado direito de (2.7), que é uma função linear, denotada por $l(\alpha)$, tem declive negativo $\mu_1 \nabla F(x_k)^T d_k$. A condição de Armijo diz que α é aceite apenas se $\phi(\alpha) \leq l(\alpha)$. Os intervalos nos quais esta condição se verifica estão ilustrados na figura 2.3. Na prática, μ_1 é escolhido como sendo bastante pequeno, digamos $\mu_1 = 10^{-4}$. Valores exageradamente

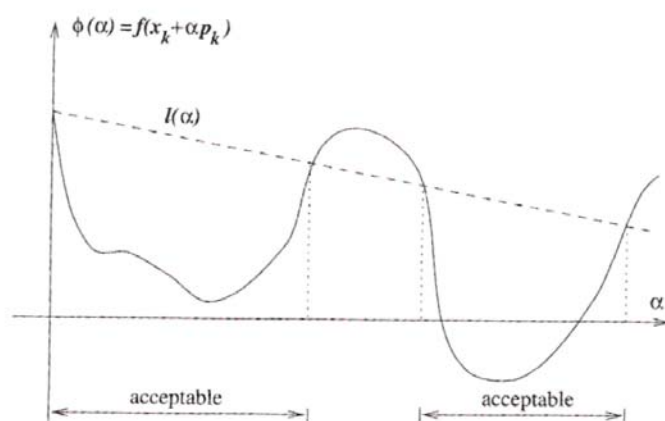


Figura 2.3: Condição de redução significativa.

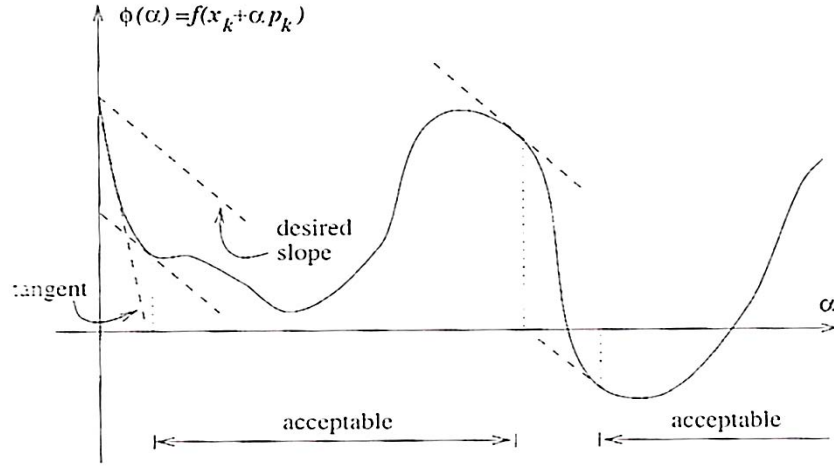


Figura 2.4: Condição de curvatura.

grandes de α não são aceites pela condição de Armijo, embora possam verificá-la valores de α muito pequenos. Para ultrapassar este inconveniente, é vantajoso adicionar outra condição para aceitação de um valor de α_k . A condição adicional, conhecida por *condição de curvatura*, é definida por

$$\underbrace{\nabla F(x_k + \alpha_k d_k)^T d_k}_{\phi'(\alpha_k)} \geq \mu_2 \underbrace{\nabla F(x_k)^T d_k}_{\phi'(0)}, \quad (2.8)$$

onde a constante $\mu_2 \in (\mu_1, 1)$, sendo μ_1 a constante usada em (2.7). Notar que o lado esquerdo é simplesmente a derivada $\phi'(\alpha_k)$. Portanto a condição de curvatura garante que o declive de $\phi(\alpha_k)$ é maior do que μ_2 vezes o gradiente $\phi'(0)$. Isto faz sentido porque se o declive $\phi'(\alpha)$ é “fortemente” negativo, tem-se a indicação que podemos reduzir F significativamente, movendo mais ao longo da direcção d_k . Por outro lado, se o declive é apenas “levemente” negativo ou mesmo positivo, é um sinal de que não podemos esperar muito mais redução de F ao longo dessa direcção, e portanto faz sentido terminar a procura unidimensional. A condição de curvatura está ilustrada na figura 2.4.

Valores típicos de μ_2 são 0.9 quando a direcção de procura d_k é obtido pelo método de Newton ou a partir de um método quasi-Newton, e 0.1 quando d_k é obtido a partir de um método dos gradientes conjugados. A condição de redução significativa juntamente com

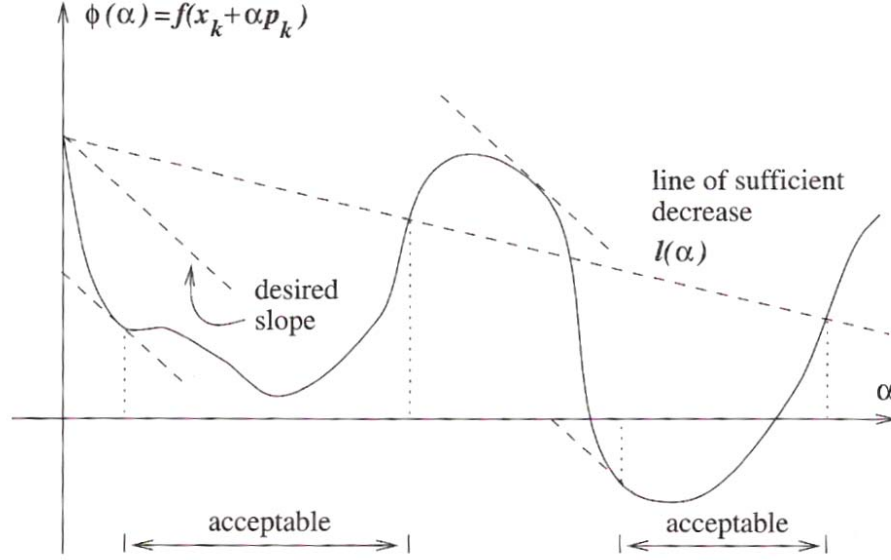


Figura 2.5: Os comprimentos do passo que satisfazem as condições de Wolfe.

a condição de curvatura são conhecidas como as *condições de Wolfe*:

$$F(x_k + \alpha_k d_k) \leq F(x_k) + \mu_1 \alpha_k \nabla F(x_k)^T d_k \quad (2.9)$$

$$\nabla F(x_k + \alpha_k d_k)^T d_k \geq \mu_2 \nabla F(x_k)^T d_k, \quad (2.10)$$

com $0 < \mu_1 < \mu_2 < 1$. Estas condições estão ilustradas na figura 2.5.

Um comprimento do passo pode satisfazer as condições de Wolfe sem estar particularmente perto de um mínimo de ϕ , como se pode ver na figura 2.5. Contudo, pode-se modificar a condição de curvatura para forçar α_k permanecer pelo menos numa vizinhança de um mínimo local ou ponto estacionário de ϕ . As *condições fortes de Wolfe* exigem que α_k satisfaça

$$F(x_k + \alpha_k d_k) \leq F(x_k) + \mu_1 \alpha_k \nabla F(x_k)^T d_k \quad (2.11)$$

$$|\nabla F(x_k + \alpha_k d_k)^T d_k| \leq \mu_2 |\nabla F(x_k)^T d_k|, \quad (2.12)$$

com $0 < \mu_1 < \mu_2 < 1$. A única diferença em relação às condições de Wolfe é que agora não se permite que a derivada $\phi'(\alpha_k)$ seja muito positiva. Portanto, exclui-se os valores de α que estão longe dos pontos estacionários de ϕ .

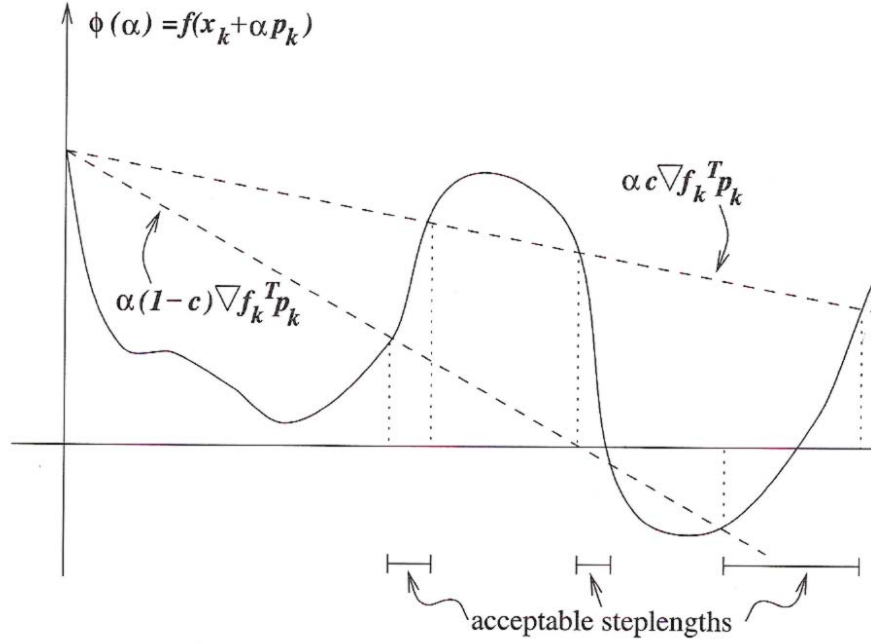


Figura 2.6: As condições de Goldstein.

As condições de Wolfe são usadas em muitos métodos de procura unidimensional, e são particularmente importantes na implementação de métodos quasi-Newton, como iremos ver na secção 2.3.5.

Condições de Goldstein Tal como as condições de Wolfe, as *condições de Goldstein* também garantem que o valor de α_k encontrado satisfaça a condição redução significativa e impedem de α_k ser muito pequeno. As condições de Goldstein são:

$$\begin{aligned} F(x_k + \alpha_k d_k) &\leq F(x_k) + \mu \alpha_k \nabla F(x_k)^T d_k \\ F(x_k + \alpha_k d_k) &\geq F(x_k) + (1 - \mu) \alpha_k \nabla F(x_k)^T d_k, \end{aligned} \quad (2.13)$$

com $0 < \mu < \frac{1}{2}$. A primeira desigualdade é a condição de redução significativa (2.7), ao passo que a segunda desigualdade é introduzida para controlar o comprimento do passo inferiormente; ver figura 2.6.

A desvantagem das condições de Goldstein versus as condições de Wolfe é que a segunda desigualdade em (2.13) pode excluir todos os minimizantes de ϕ . Contudo, as condições de Wolfe e Goldstein têm muito em comum, e suas teorias de convergência são bastante similares. As condições de Goldstein são frequentemente usadas em métodos do

tipo Newton mas não são muito apropriadas para métodos quasi-Newton, que mantêm uma aproximação definida positiva à Hessiana de F .

Condição de Armijo e redução sucessiva do passo Como já referido, o uso apenas da condição de redução significativa (2.9) não é suficiente para garantir que o algoritmo faça progresso razoável ao longo de uma dada direcção. Por vezes, o valor de α_k aceitável, por forma a originar a redução de F exigida pela condição, é pequeno de mais e o deslocamento resultante $\alpha_k d_k$ torna-se tão insignificante que o novo ponto “quase” coincide com x_k . Contudo, se o procedimento de procura unidimensional escolher os comprimentos do passo, apropriadamente, usando uma estratégia designada de *redução sucessiva do passo* (em inglês, *backtracking*), pode-se dispensar a condição adicional (2.10) e usar apenas a condição de Armijo para terminar o procedimento de procura unidimensional. Na sua forma mais básica, a redução sucessiva do passo procede-se do seguinte modo: gera-se uma sequência de valores,

$$\{\rho^j \alpha\}, \quad j = 0, 1, 2, \dots$$

a partir de um valor inicial α e com uma constante de redução ρ , os valores vão sucessivamente tornando-se mais pequenos. O valor inicial mais comum é a unidade. O primeiro elemento da sequência que verifica condição de redução significativa,

$$F(x_k + \rho^j \alpha d_k) \leq F(x_k) + \mu_1 \rho^j \alpha \nabla F(x_k)^T d_k,$$

é aceite na iteração k

$$\alpha_k \leftarrow \rho^j \alpha.$$

O factor de redução na sequência $\{\rho^j \alpha\}$ é $\rho = \frac{1}{2}$. Outros valores podem aqui ser usados, e até para o valor inicial de partida não há regra que determine qual o mais apropriado. Nos métodos de Newton e quasi-Newton, o valor inicial α é escolhido como sendo a unidade, mas pode ter valores diferentes nos outros algoritmos tais como o de descida máxima ou dos gradientes conjugados. Notar que, encontra-se sempre um comprimento do passo aceitável após um número de tentativas, porque $\rho^j \alpha$ torna-se pequeno o suficiente, que a condição de Armijo verifica-se. Na prática, quando o mínimo está muito próximo, o processo torna-se muito lento pois o valor de α_k aceitável, por forma a originar a redução de F exigida pela condição de Armijo, é pequeno demais, o que obriga a um número exagerados de cálculos da função F . Sugere-se a utilização de $\alpha_k = 1$ nas iterações em que esta situação ocorre, mesmo que o valor de F não sofra um decréscimo.

Algoritmo de procura unidimensional baseado na condição de Armijo

1. Dar $\bar{\alpha} > 0$, $\rho = \frac{1}{2}$, $\mu_1 \in (0, 1)$. Fazer $\alpha = \bar{\alpha}$
2. Se $F(x_k + \alpha d_k) > F(x_k) + \mu_1 \alpha \nabla F(x_k)^T d_k$ então
 - (a) Fazer $\alpha = \frac{\alpha}{2}$
 - (b) Se $\alpha \|d_k\|_2 \leq 10^{-8}$ então fazer $\alpha = 1$ e ir para o passo 4, senão ir para o passo 2
3. Terminar com $\alpha_k \leftarrow \alpha$.

2.3.2 Critério de paragem

Idealmente um algoritmo para optimização sem restrições deveria terminar numa estimativa da solução, x_{k+1} , que satisfizesse

$$\nabla F(x_{k+1}) = 0 \quad \text{e} \quad \nabla^2 F(x_{k+1}) \text{ semidefinida positiva.}$$

Existem duas razões para que isto não seja realístico. Primeiro, é improvável que o valor calculado do gradiente seja sempre exactamente zero devido aos erros de arredondamento, cometidos nos cálculos por computador. Segundo, mesmo que não existissem erros de arredondamento, nenhum algoritmo garante encontrar um tal ponto numa quantidade finita de tempo. Outra dificuldade, embora esta não seja tão grave, é que muitos dos algoritmos vulgarmente usados para optimização sem restrições não têm disponível a matriz Hessiana $\nabla^2 F(x_{k+1})$, de modo que não podem directamente verificar se esta matriz é semidefinida positiva. Como alternativa, podemos substituir as condições acima pelo teste, de verificar se a norma do gradiente, $\|\nabla F(x_{k+1})\|$, decresce e tende para zero, com k :

$$\|\nabla F(x_{k+1})\| \leq \varepsilon \tag{2.14}$$

onde ε é uma quantidade pequena e positiva. Um valor apropriado para ε poderá ser $\varepsilon = \sqrt{\varepsilon_{maq}}$.

Uma vez que não é possível conceber um teste de convergência perfeito para terminar um algoritmo, é comum incluir testes adicionais para serem satisfeitos, antes de um ponto x_{k+1} ser aceite, como uma aproximação ao minimizante da função F . Por exemplo, o algoritmo pode tentar garantir que as sucessões $\{x_k\}$ e $\{F(x_k)\}$ estão ambas a convergir, respectivamente para, x^* e $F(x^*)$:

- Dado que $\|x_{k+1} - x_k\|$ é uma estimativa do erro $\|x_k - x^*\|$ em x_k , um teste adicional a incluir no critério de paragem é verificar se $\|x_{k+1} - x_k\|$ se torna mais pequeno do que uma certa tolerância.

Nota: Um critério baseado apenas nesta condição não é aconselhável pois $\|x_{k+1} - x_k\|$ não é necessariamente uma sucessão monótona decrescente de k . Por exemplo, pode existir situações em que a diferença $\|x_{k+1} - x_k\|$ já é pequena, mas a variação $|F(x_{k+1}) - F(x_k)|$ ainda é grande e x_{k+1} está ainda longe de x^* .

- Como em muitos processos, o valor da função F vai reduzindo-se, de iteração para iteração, então a sucessão $\{F(x_k)\}$ é uma sucessão monótona decrescente de k . Assim, pode-se também usar como condição de convergência o facto da quantidade $|F(x_{k+1}) - F(x_k)|$ poder tomar valores muito pequenos, a partir de uma certa iteração.

Nota: Um critério também apenas baseado nesta condição não é aconselhável. Por exemplo, pode existir problemas em que a variação $|F(x_{k+1}) - F(x_k)|$ já é pequena, a alteração $\|x_{k+1} - x_k\|$ em x é, no entanto, grande e o mínimo está ainda longe de x_{k+1} .

Tendo em conta o foi dito, é possível aconselhar dois critérios de paragem para o problema de optimização não linear.

Critério de Himmelblau (proposto em Wolfe (1978)), considera as três condições:

$$\frac{\|x_{k+1} - x_k\|}{\|x_{k+1}\|} \leq \varepsilon_2, \quad \frac{|F(x_{k+1}) - F(x_k)|}{|F(x_{k+1})|} \leq \varepsilon_3$$

e

$$\|\nabla F(x_{k+1})\| \leq \varepsilon_1.$$

O segundo critério (veja-se em Gill e Murray (1976)) é baseado nas condições:

$$\|x_{k+1} - x_k\| \leq \varepsilon (1 + \|x_{k+1}\|), \quad |F(x_{k+1}) - F(x_k)| \leq \varepsilon^2 (1 + |F(x_{k+1})|)$$

e

$$\|\nabla F(x_{k+1})\| \leq \varepsilon^{\frac{1}{3}} (1 + |F(x_{k+1})|). \quad (2.15)$$

A condição (2.15) surge pelo facto de se poder modificar a função objectivo, alterando-se as unidades em que foi medida. Por exemplo, suponha-se que em vez de medir a função objectivo em termos de kilometros fosse agora medida em termos de milímetros. Isto causaria a função objectivo ser multiplicada por 10^6 e portanto causaria $\|\nabla F(x_k)\|$ ser

multiplicada por 10^6 . Esta variação da função objectiva faz com que o teste de convergência, baseado na condição (2.14), seja mais difícil de satisfazer. Para ultrapassar esta dificuldade a condição (2.14) pode ser modificada para $\|\nabla F(x_{k+1})\| \leq \varepsilon |F(x_{k+1})|$. Mas se o valor da função objectiva é zero (que é comum quando se resolve problemas de mínimos quadrados), então esta condição é impossível de satisfazer. Por tudo isto, considera-se a condição da forma $\|\nabla F(x_{k+1})\| \leq \varepsilon (1 + |F(x_{k+1})|)$.

O critério de paragem de Gill e Murray é de aplicação mais geral pois aplica-se mesmo para problemas em que o mínimo a atingir é zero.

Em ambos os critérios, os testes no $\nabla F(x_{k+1})$ e x_{k+1} baseiam-se nas normas de vectores. Se, por exemplo, $\nabla F(x_{k+1}) = (\gamma, \dots, \gamma)^T$ então

$$\|\nabla F(x_{k+1})\|_2 = \sqrt{n} |\gamma| \quad \text{e} \quad \|\nabla F(x_{k+1})\|_\infty = |\gamma|$$

onde n é o número de variáveis. Se n é grande então a norma 2 do gradiente pode ser grande mesmo se γ é pequeno. Isto pode distorcer os testes de convergência e portanto, nos problemas de grandes dimensões é aconselhável usar a norma infinito.

2.3.3 Método de Newton

O método de Newton, considera o seguinte modelo quadrático, que aproxima a função F localmente, na vizinhança de x_k :

$$Q(x_k + d) = F(x_k) + \nabla F(x_k)^T d + \frac{1}{2} d^T \nabla^2 F(x_k) d. \quad (2.16)$$

$Q(x_k + d)$ corresponde aos três primeiros termos da expansão em série de Taylor de F em torno do ponto x_k . O objectivo é minimizar Q como uma função de d . Assim, fixando x_k , a condição necessária de 1ª ordem para que $x_{k+1} = x_k + d$ seja mínimo de Q é:

$$\nabla Q(x_k + d) \equiv \nabla F(x_k) + \nabla^2 F(x_k) d = 0$$

donde se retira que d é a solução do sistema linear:

$$\nabla^2 F(x_k) d = -\nabla F(x_k). \quad (2.17)$$

Se a função F é quadrática estritamente convexa, a derivada de segunda ordem é uma matriz constante, a aproximação (2.16) é exacta e tem um único mínimo que se atinge numa só etapa a partir de um ponto qualquer x_k .

Se a função F não é quadrática, o novo ponto obtido $x_k + d$, com d calculado das equações de Newton (2.17) não é o mínimo da função F , pelo que o processo deve ser repetido iterativamente. Assim, as equações iterativos do processo são:

$$x_{k+1} = x_k + d_k$$

com

$$\nabla^2 F(x_k) d_k = -\nabla F(x_k), \quad k = 1, 2, \dots$$

onde x_1 uma aproximação inicial à solução. Estas equações definem o método de Newton, na sua implementação original e básica.

Se x_1 está suficientemente perto de um mínimo local x^* de F , com $\nabla^2 F(x^*)$ não singular, e mesmo definida positiva, então a sucessão de pontos gerada pelo método de Newton converge para x^* e a convergência é quadrática. Por vezes, não é possível obter-se uma boa estimativa inicial x_1 e o método pode não convergir.

Existem outros factores que levam o método de Newton a falhar a convergência para a solução desejada e que são:

- $\nabla^2 F(x_k)$ é definida positiva:
 - o ponto estacionário da função quadrático é um mínimo de Q .
 - d_k é uma direcção de descida em relação F , no ponto x_k .

Nesta situação, em que d_k é de descida, pode acontecer que o seu comprimento seja tal que $F(x_k + d_k) > F(x_k)$, e o novo ponto não é melhor do que o anterior, no sentido de que a função não foi reduzida. Este problema ultrapassa-se usando simultaneamente com a direcção de Newton um procedimento de procura unidimensional. Ou seja, é possível garantir a existência de um valor positivo para α_k , de tal forma que, $F(x_k + \alpha_k d_k) < F(x_k)$.

O uso de um procedimento de procura unidimensional permite também que o método de Newton possa convergir a partir de qualquer ponto inicial, x_1 .

- $\nabla^2 F(x_k)$ é definida negativa¹:
 - o ponto estacionário da função quadrática é um máximo de Q .
 - d_k não aponta no sentido descendente de F , e não é possível garantir a existência de um valor para α que verifique $F(x_k + \alpha d_k) < F(x_k)$.
- Se $\nabla^2 F(x_k)$ é indefinida:

¹Se os determinantes das submatrizes principais de uma matriz têm sinais alternadamente negativos e positivos, sendo o primeiro negativo, a matriz diz-se definida negativa.

- d_k pode não ser uma direcção de descida, pode até tornar-se aproximadamente ortogonal à direcção de descida máxima: $-\nabla F(x_k)$. Caso não seja de descida, não é possível garantir a existência de um valor para α que verifique $F(x_k + \alpha d_k) < F(x_k)$.
- $\nabla^2 F(x_k)$ é singular
 - direcção de procura d_k não é sequer definida \rightarrow o método falha.

Se a matriz Hessiana, $\nabla^2 F(x_k)$, não é definida positiva então uma estratégia possível é substituir a Hessiana por uma matriz definida positiva, quando necessário, na fórmula para a direcção de Newton (2.17). Isto garante que a direcção de procura, d_k , é uma direcção de descida em relação a F , no ponto x_k .

Algumas razões que levaram ao uso desta estratégia:

1. Se isto for feito de modo apropriado, é possível mostrar que o algoritmo resultante converge.
2. Na solução do problema de optimização $\nabla^2 F(x^*)$ será, em geral, definida positiva (é sempre semidefinida positiva). Portanto, a Hessiana será apenas substituída nos pontos que não estão perto do ponto óptimo.
3. A matriz definida positiva pode ser obtida durante um processo de factorização da matriz Hessiana.

Calcular a direcção de procura implica resolver o sistema linear:

$$\nabla^2 F(x_k) d_k = -\nabla F(x_k).$$

Se $\nabla^2 F(x_k)$ é definida positiva, então pode-se usar a factorização

$$\nabla^2 F(x_k) = L_k L_k^T$$

onde L_k é uma matriz triangular inferior. Se $\nabla^2 F(x_k)$ não é definida positiva, então durante o processo de factorização $\nabla^2 F(x_k)$ é substituída por $\nabla^2 F(x_k) + E_k$:

$$\nabla^2 F(x_k) \rightarrow \nabla^2 F(x_k) + E_k$$

onde E_k é uma matriz diagonal de tal forma que a matriz Hessiana modificada é definida positiva. Na prática, nem $\nabla^2 F(x_k) + E_k$ nem E_k são efectivamente calculados: apenas a factorização. A factorização é então usada para calcular a direcção de procura:

$$(L_k L_k^T) d_k = -\nabla F(x_k)$$

Nota: Em cada iteração ter-se-á que calcular sempre factorização da matriz $\nabla^2 F(x_k)$, mesmo que seja esta definida positiva em todo o processo iterativo. Portanto a matriz modificada, caso seja necessária, é obtida sem qualquer custo.

Esta factorização pode ser obtida recorrendo ao seguinte algoritmo da factorização modificada de Cholesky [9].

Algoritmo da factorização modificada de Cholesky de uma matriz simétrica

Seja $G \equiv \nabla^2 F(x_k)$ uma matriz real simétrica de $n \times n$ e $\delta > 0$.

1. Calcular ξ_i ($i = 1, \dots, n-1$), ξ , γ , e β do seguinte modo, respectivamente:

$$\xi_i = \max_{i+1 \leq j \leq n} \{|G_{ji}|\} \quad (i = 1, \dots, n-1)$$

$$\xi = \max_{1 \leq i \leq n-1} \{\xi_i\}$$

$$\gamma = \max_{1 \leq i \leq n} \{|G_{ii}|\}$$

$$\beta = \max \left\{ \gamma^{\frac{1}{2}}, \left(\frac{\xi}{n} \right)^{\frac{1}{2}} \right\}$$

2. Calcular \widehat{l}_j ($j = 1, \dots, n$) e θ :

$$\widehat{l}_1 = \max \left\{ \delta, |G_{11}|^{\frac{1}{2}} \right\}$$

$$\widehat{l}_j = \frac{G_{j1}}{\widehat{l}_1} \quad (j = 2, \dots, n)$$

$$\theta = \max_{2 \leq i \leq n} \left\{ |\widehat{l}_i| \right\}$$

3. Calcular l_{j1} ($j = 1, \dots, n$):

Se $\theta \leq \beta$ então

$$l_{j1} = \widehat{l}_j \quad (j = 1, \dots, n)$$

senão

$$l_{11} = \frac{\theta}{\beta} \widehat{l}_1$$

$$l_{j1} = \frac{\beta}{\theta} \widehat{l}_j \quad (j = 2, \dots, n)$$

fimSe

4. Fazer $i = 2$.

5. Calcular \widehat{l}_i :

$$\widehat{l}_i = \max \left\{ \delta, \left| G_{ii} - \sum_{m=1}^{i-1} l_{im}^2 \right|^{\frac{1}{2}} \right\}$$

6. Se $i = n$ ir para 11.

7. (senão) Calcular \widehat{l}_j ($j = i + 1, \dots, n$):

$$\widehat{l}_j = \frac{1}{\widehat{l}_i} \left\{ G_{ji} - \sum_{m=1}^{i-1} l_{jm} l_{im} \right\}$$

8. Calcular θ :

$$\theta = \max_{i+1 \leq j \leq n} \left\{ \left| \widehat{l}_j \right| \right\}$$

9. Calcular l_{ji} ($j = i, \dots, n$):

Se $\theta \leq \beta$ então

$$l_{ji} = \widehat{l}_j \quad (j = i, \dots, n)$$

senão

$$l_{ii} = \frac{\theta}{\beta} \widehat{l}_i$$

$$l_{ji} = \frac{\beta}{\theta} \widehat{l}_j \quad (j = i + 1, \dots, n)$$

fimSe

10. Fazer $i = i + 1$ e voltar para 5.

11. Calcular l_{nn} :

$$l_{nn} = \max \left\{ \delta, \left| G_{nn} - \sum_{m=1}^{n-1} l_{nm}^2 \right|^{\frac{1}{2}} \right\}$$

12. Terminar.

Algoritmo de Newton Modificado

1. Dar x_1 e calcular $\nabla F(x_1), \nabla^2 F(x_1)$. Fazer $k = 1$.
2. Factorizar a matriz: $\nabla^2 F(x_k) + E_k = L_k L_k^T$
($E_k=0$ se $\nabla^2 F(x_k)$ “suficientemente” definida positiva).
3. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares

$$(L_k L_k^T) d_k = -\nabla F(x_k).$$
4. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional.
5. Calcular $x_{k+1} = x_k + \alpha_k d_k$, $\nabla F(x_{k+1})$ e $\nabla^2 F(x_{k+1})$.
6. Se x_{k+1} satisfaz o critério de paragem, terminar $[x^* \leftarrow x_{k+1} \text{ e } F^* \leftarrow F(x_{k+1})]$.
7. (senão) Fazer $k = k + 1$ e voltar para 2.

Existem três tipos de custos associados ao método de Newton:

- requer o cálculo das segundas derivadas da função objectivo;
- resolução de um sistema linear;
- armazenamento de uma matriz quadrada de ordem n ;

A principal desvantagem deste método era o cálculo das segundas derivadas, o que significava que $O(n^2)$ expressões teriam que ser programadas para as calcular. Ter que calcular as segundas derivadas à “mão” e programá-las era muito aborrecido; e era fácil cometer-se erros, que podiam levar a que o algoritmo tivesse fraco desempenho ou mesmo falhasse.

Hoje em dia existem disponíveis técnicas de diferenciação automática. A diferenciação automática elimina o aborrecimento de se calcular as segundas derivadas à “mão”, bem como o risco de se introduzir erros nos cálculos. E portanto, actualmente, a principal desvantagem do método de Newton está ultrapassada.

Uma vez encontrada a matriz Hessiana, resolver o sistema linear custa $O(n^3)$ operações aritméticas. E a matriz é armazenada a um custo de $O(n^2)$ posições de memória. À medida que n aumenta estes custos crescem rapidamente.

Para reduzir estes custos foram propostas muitas modificações ao método de Newton, tais como:

- usar apenas informação 1ª ordem, para evitar o cálculo das segundas derivadas;
- e em problemas de grandes dimensões, também tentar reduzir os custos do cálculo da direcção de procura e do armazenamento da Hessiana.

Os algoritmos resultantes tem taxas de convergência mais baixas. E na resolução dos problemas, tendem a fazer mais iterações mas o custo, por iteração, é inferior.

2.3.4 Método das direcções de descida máxima

O método das direcções de descida máxima é o mais simples dos métodos do tipo-Newton para optimização não linear. O preço desta simplicidade é que o método é ineficiente na resolução de muitos dos problemas. O método tem utilidade a nível teórico, quer na demonstração da convergência de outros métodos quer na demonstração de limites inferiores no desempenho de algoritmos melhores.

O método das direcções de descida máxima foi inventado no século dezanove por Cauchy, cerca de duzentos anos depois do método de Newton.

Este método é muito mais simples do que o método de Newton. Não requer o cálculo das segundas derivadas, não requer que um sistema de equações lineares seja resolvido para calcular a direcção de procura, e não requer o armazenamento da matriz. Portanto, em todos os sentidos, reduz os custos do método de Newton - pelo menos os custos por iteração. Porém, tem uma taxa de convergência mais baixa do que a do método de Newton; apenas converge a uma taxa linear com uma constante que é usualmente perto de um. Por esta razão, muitas vezes converge muito lentamente, algumas vezes tão lentamente que $x_{k+1} - x_k$ é inferior à precisão aritmética do computador e o método falha. Como conclusão, mesmo pensando que os custos por iteração são baixos, o custo total de resolver um problema de optimização por este método é alto.

O método das direcções de descida máxima calcula a direcção de procura do seguinte modo

$$d_k = -\nabla F(x_k),$$

e a seguir usa um procedimento de procura unidimensional para determinar $x_{k+1} = x_k + \alpha_k d_k$. Portanto, o custo de calcular a direcção de procura é somente o custo de calcular o gradiente. Uma vez que o gradiente tem que ser calculado para testar o critério de paragem, pode-se dizer que a direcção de procura está disponível sem qualquer custo. A direcção de procura é uma direcção de descida se $\nabla F(x_k) \neq 0$, isto é, d_k é uma direcção de descida excepto quando x_k é ponto estacionário da função F .

A fórmula para a direcção de procura pode ser deduzida de duas maneiras, e ambas as deduções estão relacionadas com o método de Newton. A primeira dedução baseia-se numa aproximação “grosseira” à Hessiana. Se é usada a fórmula do método de Newton ($\nabla^2 F(x_k)d_k = -\nabla F(x_k)$) mas com a Hessiana aproximada por a matriz identidade ($\nabla^2 F(x_k) \approx I$) então é obtida a fórmula para o método das direcções de descida máxima.

A segunda dedução, e mais tradicional, baseia-se na série de Taylor e explica o nome “direcção de descida máxima”. Na dedução do método de Newton, o valor da função $F(x_k + d)$ foi aproximado pelos três primeiros termos da série de Taylor, e a direcção de procura foi obtida minimizando essa aproximação. Aqui apenas se usa os dois primeiros termos da série:

$$F(x_k + d) \approx \underbrace{F(x_k) + \nabla F(x_k)^T d}_{L(x_k + d)}.$$

A ideia intuitiva é minimizar esta aproximação linear, $L(x_k + d)$, para obter a direcção de procura; todavia esta aproximação não tem, em geral, um mínimo finito. Como alternativa, a direcção de procura é calculada minimizando uma versão escalar desta aproximação:

$$\min_{d \neq 0} \frac{d^T \nabla F(x_k)}{\|d\| \|\nabla F(x_k)\|} \equiv \cos \theta$$

Dado que $d^T \nabla F(x_k) = \|d\| \|\nabla F(x_k)\| \cos \theta$ onde θ é o ângulo entre d e $\nabla F(x_k)$. Assim, mínimo ocorre para $\theta = \pi$, o que quer dizer que d é a direcção negativa do gradiente. Portanto a solução é $d_k = -\nabla F(x_k)$.

Se o valor de α foi determinado exactamente, isto é, se α minimiza F ao longo de d ,

$$\frac{d}{d\alpha} F(x + \alpha d) = \nabla F(x + \alpha d)^T d = 0$$

então as sucessivas direcções de descida máxima definem ângulos rectos e as direcções tendem a torna-se linearmente dependentes. O método apresenta um comportamento em zigzag (ver figura 2.7) que se traduz num processo muito lento quando está já perto do mínimo.

Longe do óptimo este método torna-se mais rápido e a sua convergência, embora sendo linear, é global, isto é, não é necessário que a aproximação inicial esteja próxima do mínimo, para ser possível provar a convergência.

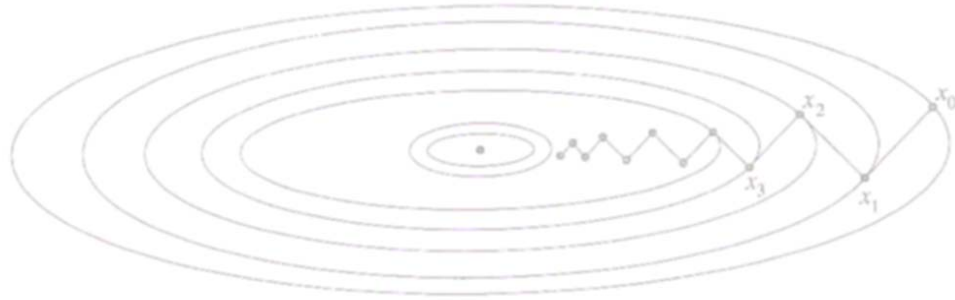


Figura 2.7: Direcções de descida máxima.

Algoritmo das direcções de descida máxima

1. Dar x_1 e calcular $\nabla F(x_1)$. Fazer $k = 1$.
2. Determinar a direcção de procura, fazendo $d_k = -\nabla F(x_k)$.
3. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional.
4. Calcular $x_{k+1} = x_k + \alpha_k d_k$, $\nabla F(x_{k+1})$.
5. Se x_{k+1} satisfaz o critério de paragem, terminar $[x^* \leftarrow x_{k+1} \text{ e } F^* \leftarrow F(x_{k+1})]$.
6. (senão) Fazer $k = k + 1$ e voltar para 2.

2.3.5 Métodos quasi-Newton

Os métodos quasi-Newton talvez sejam os métodos mais usados para optimização não linear. Estão incorporados em muitos pacotes de software, e tem sido bem sucedidos na resolução de uma grande variedade de problemas práticos.

Os métodos quasi Newton, tal como o método das direcções de descida máxima, requer apenas que o gradiente da função objectivo seja fornecido em cada iteração. Medindo a variação nos gradientes, constroem um modelo da função objectivo que é suficientemente bom para se obter convergência superlinear. É dramático o melhoramento sobre o método das direcções de descida máxima, especialmente nos problemas difíceis. Além disso, uma vez que não requerem as segundas derivadas, os métodos quasi-Newton são por vezes mais eficientes do que o método de Newton. Actualmente, os pacotes de software de optimização contêm uma variedade de algoritmos quasi-Newton para resolução de problemas de

otimização sem restrições, com restrições, e de grandes dimensões. Os métodos quasi-Newton que iremos apresentar são para problemas de pequena e média dimensão, não focaremos a sua extensão para problemas de grande dimensão.

É verdade que, o desenvolvimento de técnicas de diferenciação automática tem vindo a diminuir o recurso aos métodos quasi-Newton, mas apenas de um modo limitado. A diferenciação automática eliminou o aborrecimento de se calcular as derivadas à mão, bem como o risco de se introduzir erros nos cálculos. Apesar disso, os métodos quasi-Newton continuam competitivos em muitos tipos de problemas.

Métodos quasi-Newton

Existe uma grande variedade de métodos quasi-Newton mas todos baseiam-se na aproximação da matriz Hessiana $\nabla^2 F(x_k)$ por uma matriz B_k que está disponível a menor custo. Por conseguinte, a direcção de procura é obtida resolvendo

$$B_k d_k = -\nabla F(x_k),$$

que é, as equação de Newton (2.17) mas com a matriz Hessiana substituída por B_k . Se a matriz B_k é definida positiva então isto é equivalente a minimizar modelo quadrático convexo, relativamente a d ,

$$\underset{d}{\text{minimizar}} \quad \underbrace{F(x_k) + \nabla F(x_k)^T d + \frac{1}{2} d^T B_k d}_{m_k(x_k+d)}.$$

Os diversos métodos quasi-Newton diferem da escolha da matriz B_k .

Vantagens:

- a aproximação B_k pode ser obtida usando apenas a informação das primeiras derivadas;
- a direcção de procura pode ser calculada usando apenas $O(n^2)$ operações (versus $O(n^3)$ operações do método de Newton).

Desvantagens (mas menores):

- os métodos não convergem quadraticamente, mas podem convergir superlinearmente. A nível da precisão aritmética do computador, não existe muito diferença prática entre estas duas taxas de convergência.
- os métodos quasi-Newton também requerem o armazenamento da matriz B_k .

Condição Secante

Em vez de calcular B_k em cada iteração, a estratégia consiste em actualizá-la dum modo simples, tendo em conta a medida de curvatura durante os passos mais recentes.

Supor que, usando um procedimento de procura unidimensional, gerou-se um novo iterando $x_{k+1} = x_k + \alpha_k d_k$ e que se deseja construir um novo modelo quadrático, que aproxima a função F , localmente, na vizinhança de x_{k+1} , da forma

$$m_{k+1}(x_{k+1} + d) = F(x_{k+1}) + \nabla F(x_{k+1})^T d + \frac{1}{2} d^T B_{k+1} d.$$

Questão: Quais as condições que se devem impôr sobre aproximação B_{k+1} , com base no conhecimento ganho durante o último passo, por forma a que B_{k+1} simule o efeito da matriz Hessiana ao longo da direcção d_k .

Resposta: Duas condições razoáveis são que o gradiente de m_{k+1} ,

$$\nabla m_{k+1}(x_{k+1} + d) = \nabla F(x_{k+1}) + B_{k+1} d,$$

deve coincidir com o gradiente da função objectivo F nos dois últimos iterandos x_k e x_{k+1} :

$$\nabla m_{k+1}(x_k) \equiv \nabla F(x_k)$$

$$\nabla m_{k+1}(x_{k+1}) \equiv \nabla F(x_{k+1})$$

A segunda condição encontra-se automaticamente satisfeita pois

$$\nabla m_{k+1}(x_{k+1}) = \nabla m_{k+1}(x_{k+1} + 0) = \nabla F(x_{k+1}).$$

A primeira condição pode ser escrita matematicamente como

$$\nabla m_{k+1}(x_k) = \nabla m_{k+1}(x_{k+1} - \alpha_k d_k) = \nabla F(x_{k+1}) - \alpha_k B_{k+1} d_k \equiv \nabla F(x_k).$$

Rearranjando, obtém-se

$$B_{k+1} \alpha_k d_k = \nabla F(x_{k+1}) - \nabla F(x_k). \quad (2.18)$$

Para simplificar a notação definem-se os vectores

$$s_k = \alpha_k d_k = x_{k+1} - x_k, \quad y_k = \nabla F(x_{k+1}) - \nabla F(x_k) \quad (2.19)$$

de modo que (2.18) torna-se

$$B_{k+1} s_k = y_k. \quad (2.20)$$

Esta fórmula é conhecida por condição secante. Para um problema de dimensão n esta condição representa um conjunto de n equações que tem que ser verificadas por B_{k+1} . Dado que a matriz B_{k+1} tem n^2 elementos, então esta condição por si só não é suficiente para determinar B_{k+1} de modo único. Tem que ser impostas condições adicionais, para especificar um determinado método quasi-Newton.

Todos os métodos quasi-Newton aqui abordados têm a forma:

$$B_{k+1} = B_k + C_k, \quad k = 1, 2, \dots$$

A matriz C_k representa uma “actualização” para a “velha” aproximação B_k , da e que uma fórmula para uma aproximação quasi-Newton é muitas vezes referida como uma fórmula de actualização.

Existem várias escolhas possíveis para as matrizes C_k . Podem ser seleccionadas por forma a serem:

1. *matrizes de característica um*

$$(C = auu^T, \text{ onde } a \text{ é um escalar e } u \text{ é um vector})$$

2. *matrizes de característica dois*

$$(C = auu^T + bvv^T, \text{ onde } a \text{ e } b \text{ são escalares e } u \text{ e } v \text{ são vectores})$$

3. *matrizes de norma mínima*

(para determinar B_{k+1} de modo único, impõem a condição que, entre todas as matrizes B que satisfazem a condição secante e determinadas condições adicionais, B_{k+1} é, a que se encontra mais próxima da matriz actual B_k :

$$\min_B \|B - B_k\| \tag{2.21}$$

$$\text{s.a } Bs_k = y_k, \quad [+ \text{ condições adicionais sobre } B]$$

Pode-se usar diversas normas matriciais em (2.21), cada norma matricial dará origem a um método quasi-Newton diferente.)

Uma vez que o critério 3. origina as mesmas matrizes do que os dois primeiros, apenas são referidas fórmulas que satisfazem 1. ou 2.

Como referido, obtém-se uma variedade de métodos quasi-Newton consoante as condições adicionais impostas na aproximação B_{k+1} . Essas condições são, geralmente, propriedades da matriz Hessiana que gostaríamos que a aproximação B_{k+1} também possuí-se.

Fórmulas de actualização quasi-Newton

- Como matriz Hessiana é simétrica, também deve ser simétrica a aproximação B_k . Por exemplo, a seguinte fórmula de actualização quasi-Newton

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

preserva a simetria, porque B_{k+1} é simétrica se B_k é. É chamada a fórmula de actualização simétrica de característica-um (SR1). Esta é a única fórmula de actualização de característica-um que preserva a simetria (ver Lema 11.4 em [7]).

- A simetria não é única propriedade que pode ser imposta. Uma vez que a matriz Hessiana na solução x^* será, normalmente, definida positiva (será sempre semidefinida positiva), é razoável impor que as matrizes B_k sejam também definidas positivas. Neste caso, isto garante que o método quasi-Newton corresponde à minimização de um modelo quadrático convexo da função não linear F , e que a direcção de procura é uma direcção de descida.

Não existe nenhuma fórmula de característica-um que preserve tanto a simetria como a positividade das matrizes de aproximação. Contudo, existe uma infinidade de fórmulas de característica-dois que geram as matrizes de aproximação simétricas e definidas positivas. De todas, a mais usada e considerada a mais eficiente, é a fórmula de actualização BFGS

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \quad (2.22)$$

Facilmente se verifica que $B_{k+1} s_k = y_k$. Verificar que B_{k+1} é definida positiva não é tão fácil.

Lema 2.1 *Seja B_k uma matriz simétrica e definida positiva, e assume-se que B_{k+1} é obtida a partir de B_k usando a fórmula de actualização BFGS. Então B_{k+1} é definida positiva se e só se $y_k^T s_k > 0$.*

Prova. Prova. Veja-se no Lema 11.3 em [7]. ■

A nova matriz B_{k+1} apenas será definida positiva se $y_k^T s_k > 0$. Para garantir esta propriedade, basta impor uma das condições de Wolfe, no procedimento de procura unidimensional. Vejamos esta demonstração:

Prova. Multiplicando a condição de Wolfe (2.10), por α_k e usando (2.19) obtém-se:

$$\nabla F(x_{k+1})^T s_k \geq \mu_2 \nabla F(x_k)^T s_k$$

Usando (2.19) e (??) tem-se

$$y_k^T s_k = (\nabla F(x_{k+1}) - \nabla F(x_k))^T s_k \geq \mu_2 \nabla F(x_k)^T s_k - \nabla F(x_k)^T s_k = (\mu_2 - 1) \nabla F(x_k)^T \alpha_k d_k.$$

Uma vez que $\mu_2 < 1$ e d_k é uma direcção de descida, então o termo do lado direito é positivo, e condição de curvatura verifica-se. ■

A fórmula BFGS tem o nome das quatro pessoas que a desenvolveram: Broyden, Fletcher, Goldfarb e Shanno. Existe uma classe de fórmulas de actualização que preservam a positividade, dada pela seguinte fórmula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \zeta (s_k^T B_k s_k) u_k u_k^T,$$

onde ζ é um escalar e

$$u_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}.$$

Esta classe de fórmulas de actualização é muitas referida como sendo a classe de Broyden.

Notar que, obtém-se fórmula de actualização BFGS fazendo $\zeta = 0$. Tal como a actualização BFGS, a positividade das matrizes de aproximação é preservada se e só se $s_k^T y_k > 0$. Quando $\zeta = 1$, obtém-se a fórmula de actualização DFP

$$B_{k+1} = B_k + \left(1 + \frac{s_k^T B_k s_k}{y_k^T s_k}\right) \frac{y_k y_k^T}{y_k^T s_k} - \left(\frac{B_k s_k y_k^T}{y_k^T s_k} + \frac{y_k s_k^T B_k}{y_k^T s_k}\right)$$

A fórmula DFP também é designada pelo nome das três pessoas que a desenvolveram: Davidon, Fletcher e Powell.

Algoritmo quasi-Newton: BFGS

1. Dar x_1 e dar uma matriz inicial de aproximação B_1 (por ex. $B_1=I$). Calcular $\nabla F(x_1)$. Fazer $k = 1$
2. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares

$$B_k d_k = -\nabla F(x_k)$$

3. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional
4. Calcular $x_{k+1} = x_k + \alpha_k d_k$, $\nabla F(x_{k+1})$, $s_k = x_{k+1} - x_k$, $y_k = \nabla F(x_{k+1}) - \nabla F(x_k)$
5. Actualizar B_{k+1} usando a fórmula de actualização BFGS, (2.22)
6. Se x_{k+1} satisfaz o critério de paragem, terminar [$x^* \leftarrow x_{k+1}$ e $F^* \leftarrow F(x_{k+1})$]
7. (senão) Fazer $k = k + 1$ e voltar para 2

Actualização das matrizes inversas

Usando a fórmula de Sherman-Morrison-Woodbury (Lema 8.3.1 de [4])

$$(M + uv^T)^{-1} = M^{-1} - \frac{M^{-1}uv^T M^{-1}}{1 + v^T M^{-1}u}$$

para vectores u e v de R^n e matrizes quadradas não singulares M , é possível determinar as inversas das matrizes B_{k+1} , e que denotaremos por $H_{k+1} = B_{k+1}^{-1}$.

Com as fórmulas de actualização para as inversas, a direcção de procura pode ser calculada através de

$$d_k = -H_k \nabla F(x_k),$$

sem haver necessidade de resolver o sistema de equações lineares. Portanto, a direcção de procura é calculada usando apenas $O(n^2)$ operações (versus $O(n^3)$ operações do método de Newton).

Inversa da matriz B_{k+1} da fórmula BFGS:

$$H_{k+1} = H_k + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right) \frac{s_k s_k^T}{y_k^T s_k} - \left(\frac{H_k y_k s_k^T}{y_k^T s_k} + \frac{s_k y_k^T H_k}{y_k^T s_k}\right)$$

Inversa da matriz B_{k+1} da fórmula DFP:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}$$

Estratégia de skipping

O desempenho do método quasi-Newton BFGS pode degrada-se se o procedimento de procura unidimensional não é baseado numa das condições de Wolfe. Por exemplo, caso o procedimento de procura unidimensional seja apenas baseado na condição de Armijo e redução sucessiva do passo, não há garantia que condição de curvatura $y_k^T s_k > 0$ seja satisfeita pelo comprimento do passo escolhido. Nos casos em que a condição de curvatura falha, usa-se $|y_k^T s_k|$. É também utilizada uma estratégia de *skipping* que consiste em não aplicar a fórmula de actualização BFGS fazendo $B_{k+1} = B_k$, para ultrapassar os inconvenientes que surgem quando os escalares que aparecem nos denominadores da referida fórmula de actualização, $y_k^T s_k$ e $s_k^T B_k s_k$, se tornam muito pequenos. Toda esta estratégia será designada simplesmente por estratégia de *skipping*.

estratégia skipping:

- Se $|y_k^T s_k| \leq \gamma$ ou $s_k^T B_k s_k \leq \gamma$ então $B_{k+1} = B_k$, onde γ designa o parâmetro de *skipping*.

2.4 Exercícios

1. Dada a função $F : R \longrightarrow R$ definida por

$$F(x) = x^2 - x$$

calcule o seu mínimo usando o método de procura de Davis, Swann e Campey (DSC), baseado na interpolação quadrática. O processo iterativo deve ser iniciado com $x_1 = 2$. Considere $\Delta = 1$, $M = 0.5$ e $\varepsilon = 0.5$.

2. Dada a função $F : R \longrightarrow R$ definida por

$$F(x) = \frac{10^4 x}{(x + 20)^2}$$

calcule o valor de x que maximiza esta função, utilizando o método de DSC baseado na interpolação quadrática. Utilize como valor inicial $x_1 = 15$, $\Delta = 2$, $M = 0.5$ e $\varepsilon = 0.5$.

3. Considere a função $F : R \longrightarrow R$ definida por

$$F(x) \begin{cases} (x-1)^2 & \text{se } x \notin [0, 2] \\ 1 & \text{se } x \in [0, 2]. \end{cases}$$

Use o método de DSC baseado na interpolação quadrática para calcular o seu mínimo. O processo iterativo deve ser iniciado com $x_1 = 4$. Considere $\Delta = 0.5$, $M = 0.5$ e $\varepsilon = 0.5$.

4. Dada a função $F : R^2 \longrightarrow R$ definida por

$$F(x_1, x_2) = -x_1^2 - 6x_2^2$$

calcule o seu máximo usando o algoritmo de Newton. O processo iterativo deve ser iniciado com o ponto $(1, -1)$ e deve terminar quando o critério de paragem baseado na condição $\|\nabla F(x)\| \leq \varepsilon(1 + |F(x)|)$ for verificado para $\varepsilon = 10^{-6}$. Deve, também, implementar o algoritmo DSC baseado na interpolação quadrática, para calcular o comprimento do passo, α , em cada iteração. Nesta fase, considere como valor de partida $\alpha_1 = 0$ e $\Delta = 1$.

5. Dada a função $F : R^2 \longrightarrow R$ definida por

$$F(x_1, x_2) = x_1 x_2^2 + (2 - x_1)^2$$

calcule o seu mínimo usando o algoritmo de Newton modificado. O processo iterativo deve ser iniciado com o ponto $(1, 1)$ e deve terminar quando o critério de paragem de Himmelblau for verificado para $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0.1$ ou itMax=3. No processo de factorização da matriz Hessiana considere $\delta = 1$. Deve, também, implementar o algoritmo baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$.

6. Dada a função $F : R^2 \longrightarrow R$ definida por

$$F(x_1, x_2) = x_1^2 + 2x_2^2$$

calcule o seu mínimo usando o algoritmo de descida máxima. O processo iterativo deve ser iniciado com o ponto $(1, 1)$ e deve terminar quando o critério de paragem baseado na condição $\|\nabla F(x)\| \leq \varepsilon$ for verificado para $\varepsilon = 0.1$. Deve, também, implementar o algoritmo baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$.

7. Dada a função $F : R^2 \longrightarrow R$ definida por

$$F(x_1, x_2) = x_1^2 + x_2^2 - x_1x_2$$

calcule o seu mínimo usando o algoritmo quasi-Newton, na versão BFGS. O processo iterativo deve ser iniciado com o ponto $(1, 0)$, $H_0 = I$ e deve terminar quando a condição $\|\nabla F(x)\| \leq \varepsilon$ for verificado para $\varepsilon = 0.02$. Deve, também, implementar o algoritmo baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$.

8. O lucro, em milhares de euros, da colocação de um sistema eléctrico é dado por

$$G(x_1, x_2) = 20x_1 + 26x_2 + 4x_1x_2 - 4x_1^2 - 3x_2^2$$

em que x_1 e x_2 designam, respectivamente, o custo da mão de obra e do material. Determine o lucro máximo usando o algoritmo quasi-Newton, na versão BFGS. O processo iterativo deve ser iniciado com o ponto $(0, 0)$, $H_0 = I$ e deve terminar quando a condição $\|\nabla F(x)\| \leq \varepsilon$ for verificado para $\varepsilon = 0.5$ ou itMax=2. Deve, também, implementar o algoritmo baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$.

9. Escreva um programa em MatLab para minimizar uma função mutidimensional para cada um dos seguintes algoritmos quasi-Newton: BFGS, DFP e SR1. Use $H_0 = I$ como aproximação inicial à Hessiana. Inclua o seguinte:

- (a) Use a estratégia de redução sucessiva do passo apresentada na subsecção 2.3.1 com $\mu_1 = 0.1$.
- (b) Utilize a estratégia de *skipping* apresentada com parâmetro de recomeço $\gamma = 10^{-12}$.
- (c) Aceite x como uma solução do problema de otimização se $\|\nabla F(x)\| \leq \varepsilon (1 + |Fx|)$, ou se o número de iterações excede o limite máximo pré-estabelecido (itMax). Use $\varepsilon = 10^{-8}$ e itMax=200.
- (d) Escreva o valor inicial, e em cada iteração escreva a direcção de procura, o comprimento do passo α e a nova aproximação à solução x_{k+1} . Indique se não foi encontrada nenhuma solução ao fim do limite máximo de iterações pré-estabelecido.
- (e) Teste os algoritmos com os seguintes problemas teste:

$$F_0(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \quad x_0 = (-1.2, 1)^T$$

$$F_1(x) = 4x_1^2 + 2x_2^2 + 4x_1x_2 - 3x_1, \quad x_0 = (2, 2)^T$$

$$F_2(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2, \quad x_0 = (0, 0)^T$$

$$F_3(x) = (x_1 + x_2)^4 + x_2^2, \quad x_0 = (2, -2)^T$$

$$F_4(x) = \frac{1}{2} (2x_1^2 + 3x_2^2 + 4x_3^2) + 8x_1 + 9x_2 + 8x_3, \quad x_0 = (0, 0, 0)^T$$

$$F_5(x) = \frac{1}{2} (5x_1^2 + 7x_2^2 + 9x_3^2 + 4x_1x_2 + 2x_1x_3 + 6x_2x_3) + 9x_1 + 8x_3, \\ x_0 = (0, 0, 0)^T$$

$$F_6(x) = x_1^2 + x_2^2 + x_3^2, \quad x_0 = (1, 1, 1)^T$$

$$F_7(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + c(x_1^2 + x_2^2 - 0.25)^2, \quad x_0 = (1, -1)^T$$

(teste três valores para o parâmetro c : $c = 1$, $c = 10$ e $c = 100$)

Capítulo 3

Problemas de mínimos quadrados não linear

Um problema de mínimos quadrados não linear é um problema de minimização sem restrições da forma

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad F(x) = \frac{1}{2} \sum_{j=1}^m (f_j(x))^2, \quad (3.1)$$

onde a função objectivo, F , está definida em termos de funções auxiliares $\{f_j\}$. É chamado “mínimos quadrados” porque minimiza a soma dos quadrados dessas funções. Assume-se que, cada f_j é uma função não linear duas vezes continuamente diferenciável de \mathbb{R}^n em \mathbb{R} , e que $m \geq n$.

No contexto do ajuste de dados, as funções auxiliares $\{f_j\}$ não são arbitrárias. Num problema de ajuste de dados, cada função f_j corresponde a um resíduo.

Esta área tem despertado um grande interesse da comunidade científica, principalmente devido à sua aplicabilidade em muitos problemas práticos. Quando se pretende estimar os parâmetros de um determinado modelo matemático que descreve o comportamento de um dado sistema químico, físico, financeiro ou económico, usa-se uma formulação do tipo (3.1) para medir a discrepância entre os valores dados pelo modelo e os obtidos experimentalmente. Minimizando a função F , seleccionam-se os valores para os parâmetros que melhor ajustam o modelo aos dados. A investigação que tem sido desenvolvida nas áreas da análise numérica e optimização tem originado algoritmos de minimização robustos e eficientes que exploram a estrutura especial de F e das suas derivadas.

Reunindo as componentes individuais f_j de F num vector resíduo $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ definido por

$$f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \quad (3.2)$$

consegue-se perceber porque é que a forma especial de F torna os problemas de mínimos quadrados mais fáceis de resolver do que os problemas de minimização gerais. Usando esta notação, F pode ser reescrita na forma

$$\underset{x \in \mathbb{R}^n}{\text{minimizar}} \quad F(x) = \frac{1}{2} \sum_{j=1}^m (f_j(x))^2 \equiv \frac{1}{2} f(x)^T f(x).$$

As derivadas de F podem ser expressas em termos do Jacobiano de $f(x)$, que é a matriz $m \times n$ das primeiras derivadas parciais definida por

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}. \quad (3.3)$$

Tem-se, então

$$\nabla F(x) = \sum_{j=1}^m f_j(x) \nabla f_j(x) = J(x)^T f(x), \quad (3.4)$$

$$\begin{aligned} \nabla^2 F(x) &= \sum_{j=1}^m \nabla f_j(x) \nabla f_j(x)^T + \sum_{j=1}^m f_j(x) \nabla^2 f_j(x) \\ &= J(x)^T J(x) + \sum_{j=1}^m f_j(x) \nabla^2 f_j(x). \end{aligned} \quad (3.5)$$

Em muitas aplicações, é possível calcular as primeiras derivadas parciais que tornam explícita a matriz do Jacobiano $J(x)$. Por sua vez, estas podem ser usadas para calcular o gradiente $\nabla F(x)$ como definido na fórmula (3.4). Contudo, o que caracteriza os problemas de mínimos quadrados é que, conhecendo o Jacobiano, se pode calcular a primeira parte da Hessiana $\nabla^2 F(x)$ sem qualquer custo. Além disso, o termo $J(x)^T J(x)$ é muitas vezes mais importante do que o segundo termo de (3.5), devido a quasi-linearidade do modelo perto da solução (que é, $\nabla^2 f_j$ pequeno) ou ao facto dos resíduos serem pequenos (que é, f_j pequeno). Muitos algoritmos de mínimos quadrados não lineares exploram estas propriedades da estrutura da Hessiana.

Em algumas aplicações de grande dimensão, é mais prático calcular o gradiente $\nabla F(x)$ através de técnicas de diferenciação computacionais [8, Capítulo 7] do que calcular as primeiras derivadas parciais (3.3) separadamente e formar $J(x)^T f(x)$.

Os métodos que vão ser discutidos nesta secção não se aplicam a esta situação.

O problema (3.1) é classificado como sendo de grande dimensão se o número de parâmetros n e o número de resíduos m são ambos elevados. Se n é pequeno e m é

grande, então os algoritmos que vão ser apresentados podem também ser aplicados directamente, possivelmente com algumas modificações tendo em conta as particularidades do problema. Nesta situação, não se deve formar a matriz do Jacobiano, $J(x)$, pois pode ser demasiado grande para ser armazenada. Em vez disso, calcula-se directamente $J(x)^T J(x)$ e o vector gradiente $J(x)^T f(x)$. Isto é, à medida que $f_j(x)$ e $\nabla f_j(x)$ vão sendo calculados para $j = 1, 2, \dots, m$ vão se construindo os respectivos somatórios

$$\begin{aligned}\nabla F(x) &= \sum_{j=1}^m f_j(x) \nabla f_j(x) \\ J(x)^T J(x) &= \sum_{j=1}^m \nabla f_j(x) \nabla f_j(x)^T.\end{aligned}$$

3.1 Classificação do problema relativamente aos resíduos

O problema de mínimos quadrados pode ser classificado de acordo com o valor que o vector das funções residuais f (ou F) toma na solução x^* , em:

- problema de resíduos nulos: se $f(x^*)$ é igual a zero. Nas aplicações que advêm do ajuste de um modelo aos dados, significa que o ajuste é perfeito.
- problema de pequenos resíduos: se $F(x^*)$ não é exactamente zero, mas pode atingir valores muito próximos de zero. Nestes casos, diz-se que o ajuste é bom e o vector f tem elementos muito pequenos.
- problema de grandes resíduos: se $\|f(x^*)\|$ é grande. É preciso algum cuidado com a quantidade que se usa para definir um problema de grandes resíduos, dado que, um valor grande de $F(x)$ pode dever-se ao escalonamento da função $F(x)$ e não à dificuldade no ajuste. Define-se um problema como sendo de grandes resíduos se a quantidade $F(x)/\|J(x)^T J(x)\|$ é grande.

3.2 Métodos para problemas de mínimos quadrados não lineares

Os métodos para a resolução do problema de mínimos quadrados não lineares, (3.1), exploram a estrutura do gradiente e da Hessiana de F :

$$\begin{aligned}\nabla F(x) &= J(x)^T f(x), \\ \nabla^2 F(x) &= J(x)^T J(x) + \sum_{j=1}^m f_j(x) \nabla^2 f_j(x).\end{aligned}$$

Seja x^* a solução do problema de mínimos quadrados, e suponhamos que $F(x^*) = 0$. Então $f_j(x^*) = 0$ para todo j , o que indica que todos os resíduos são zero e que o ajuste do modelo aos dados é perfeito. Como consequência, $f(x^*) = 0$ e portanto $\nabla F(x^*) = 0$, o que confirma que a condição necessária de 1ª ordem se verifica. Também se tem

$$\nabla^2 F(x^*) = J(x^*)^T J(x^*)$$

de modo que a Hessiana na solução é semidefinida positiva, como esperado. Se $J(x^*)$ é uma matriz de característica completa então $\nabla^2 F(x^*)$ é definida positiva.

Se $f(x^*) = 0$ então é razoável esperar-se que $f(x) \approx 0$ para $x \approx x^*$, o que implica que

$$\nabla^2 F(x) = J(x)^T J(x) + \sum_{j=1}^m f_j(x) \nabla^2 f_j(x) \approx J(x)^T J(x). \quad (3.6)$$

Esta última fórmula apenas envolve as primeiras derivadas das funções $\{f_j\}$ e sugere como encontrar uma aproximação à matriz Hessiana, usando apenas as primeiras derivadas, pelo menos nos casos em que o ajuste do modelo aos dados é bom. Esta ideia é a base de um número de métodos especializados para ajuste de dados por mínimos quadrados não lineares.

3.2.1 Método de Gauss-Newton

O mais simples desses métodos, chamado o método de Gauss-Newton, usa directamente a aproximação (3.6). E calcula uma direcção de procura usando a fórmula do método de Newton

$$\nabla^2 F(x_k) d_k = -\nabla F(x_k),$$

mas substitui a matriz Hessiana pela aproximação (3.6)

$$J^T(x_k) J(x_k) d_k = -J^T(x_k) f(x_k).$$

Algoritmo Gauss-Newton

1. Dar x_1 e calcular $J(x_1)$, $f(x_1)$, $\nabla F(x_1) = J(x_1)^T f(x_1)$. Fazer $k = 1$
2. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares

$$(J^T(x_k) J(x_k)) d_k = -\nabla F(x_k)$$

3. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional

4. Calcular $x_{k+1} = x_k + \alpha_k d_k$
5. Se x_{k+1} satisfaz o critério de paragem, terminar $[x^* \leftarrow x_{k+1} \text{ e } F^* \leftarrow F(x_{k+1})]$; senão ir para 6
6. Calcular $J(x_{k+1}), f(x_{k+1}), \nabla F(x_{k+1}) = J(x_{k+1})^T f(x_{k+1})$. Fazer $k = k + 1$ e voltar para 2

Vantagens:

- Como despreza o segundo termo da matriz Hessiana, as Hessianas individuais $\nabla^2 f_j(x)$, $j = 1, 2, \dots, n$, das funções residuais não são calculadas. E dado que o Jacobiano é necessário ao cálculo de vector gradiente, a aproximação $J^T(x_k)J(x_k)$ à Hessiana é obtida a um custo muito reduzido.
- O termo $J(x_k)^T J(x_k)$ é muitas vezes mais importante do que o segundo termo de (3.6), de modo que o método de Gauss-Newton dá um desempenho bastante similar ao do método de Newton, mesmo quando o segundo termo, $\sum_{j=1}^m f_j(x) \nabla^2 f_j(x)$, é omitido. Uma condição suficiente para que o primeiro termo em (3.6) domine o segundo é que o tamanho de cada termo de segunda ordem (que é, $\|f_j(x)\| \|\nabla^2 f_j(x)\|$) seja significativamente mais pequeno do que os valores próprios de $J^T J$.
Isto acontece, por exemplo, em problemas de pequenos resíduos (que é, f_j pequeno) ou quando as funções residuais são quase lineares (de modo que $\|\nabla^2 f_j(x)\|$ é pequeno). Na prática, muitos problemas de mínimos quadrados tem pequenos resíduos na solução, pelo que é muitas vezes observada a convergência local rápida do método de Gauss-Newton.
- Sempre que $J(x_k)$ tem característica completa e o gradiente $\nabla F(x_k)$ é não nulo, a direcção de procura d_k é uma direcção de descida para F , e por conseguinte uma direcção apropriada para um procedimento de procura unidimensional.

Desvantagens:

- O método de Gauss-Newton apresenta fraco desempenho quando os resíduos na solução são muito grandes (que é, quando o modelo não ajusta bem os dados), ou em que funções residuais são fortemente não lineares. O significa que nestes casos, o método de Gauss-Newton usa uma fraca aproximação à Hessiana de F .
- Se $J(x_k)$ não tem característica completa, então $J^T(x_k)J(x_k)$ é singular e a direcção de procura d_k não é sequer definida \rightarrow o método falha.

A análise de convergência deste método estabelece:

- convergência q-linear local rápida \rightarrow em problemas de pequenos resíduos ou em que as funções f_j são quase lineares;
- convergência q-linear local lenta \rightarrow em problemas de grandes resíduos ou em que as funções f_j são fortemente lineares;
- convergência q-quadrática local \rightarrow apenas em problemas lineares ou de resíduos nulos.

Uma vez que, nos algoritmos de mínimos quadrados não lineares, o cálculo da matriz do Jacobiano, $J(x)$, é necessário para o vector gradiente e pode ser feito analítica ou numericamente de forma expedita. A parte $J(x)^T J(x)$ de $\nabla^2 F(x)$ fica disponível a um custo muito reduzido, sendo muitas vezes a parte dominante de $\nabla^2 F(x)$.

Assim, muitos dos métodos para mínimos quadrados não lineares, aproximam apenas o segundo termo da matriz Hessiana

$$\sum_{j=1}^m f_j(x) \nabla^2 f_j(x).$$

A mais velha e mais simples destas aproximações é dada pelo método de Levenberg-Marquardt.

3.2.2 Método de Levenberg-Marquardt

O método de Levenberg-Marquardt simplesmente aproxima o segundo termo da matriz Hessiana por uma matriz diagonal

$$\sum_{j=1}^m f_j(x) \nabla^2 f_j(x) \approx \lambda I,$$

onde $\lambda \geq 0$ é um escalar. A direcção de procura é obtida resolvendo o sistema linear

$$(J^T(x_k) J(x_k) + \lambda I) d_k = -J^T(x_k) f(x_k).$$

As propriedades de convergência local deste método são idênticas às do método de Gauss-Newton. A vantagem deste método relativamente ao método de Gauss-Newton é que ultrapassa uma possível singularidade de $J^T J$.

Algoritmo Levenberg-Marquardt

1. Dar x_1 e um escalar positivo λ . Calcular $J(x_1)$, $f(x_1)$, $\nabla F(x_1) = J(x_1)^T f(x_1)$. Fazer $k = 1$
2. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares

$$(J^T(x_k) J(x_k) + \lambda I) d_k = -\nabla F(x_k)$$
3. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional
4. Calcular $x_{k+1} = x_k + \alpha_k d_k$
5. Se x_{k+1} satisfaz o critério de paragem, terminar $[x^* \leftarrow x_{k+1} \text{ e } F^* \leftarrow F(x_{k+1})]$; senão ir para 6
6. Calcular $J(x_{k+1})$, $f(x_{k+1})$, $\nabla F(x_{k+1}) = J(x_{k+1})^T f(x_{k+1})$. Fazer $k = k + 1$ e voltar para 2

3.2.3 Métodos quasi-Newton estruturados

Uma outra classe de métodos para a resolução do problema (3.1) é a dos métodos quasi-Newton. Uma aplicação directa dos métodos quasi-Newton ao problema de mínimos quadrados não lineares não traz vantagem pois aproximam toda matriz Hessiana da função objectivo, F .

Dado que, nos algoritmos de mínimos quadrados não lineares, o termo $J(x)^T J(x)$ de $\nabla^2 F(x)$ está sempre disponível a um custo muito reduzido, e é muitas vezes a parte dominante de $\nabla^2 F(x)$, os métodos quasi-Newton que não utilizam esta informação têm-se mostrado pouco eficientes na prática.

Assim parece mais vantajoso, combinando as ideias dos métodos Gauss-Newton e Quasi-Newton, aproximar apenas a segunda parte da matriz Hessiana.

Neste contexto, a direcção de procura, d_k , é calculada resolvendo o sistema de equações lineares

$$(J^T(x_k) J(x_k) + A_k) d_k = -J^T(x_k) f(x_k) \quad (3.7)$$

onde a matriz $n \times n$ A_k é uma aproximação à segunda parte da matriz Hessiana de F , na iteração k , sendo actualizada por fórmulas actualizadoras do tipo quasi-Newton. A matriz A_k é actualizada de modo que a nova matriz A_{k+1} satisfaz a **condição secante**

$$A_{k+1} s_k = u_k, \quad u_k = y_k - J^T(x_{k+1}) J(x_{k+1}) s_k \quad (3.8)$$

ou

$$A_{k+1} s_k = v_k, \quad v_k = (J(x_{k+1}) - J(x_k))^T f(x_{k+1}) \quad (3.9)$$

onde

$$s_k = x_{k+1} - x_k \quad \text{e} \quad y_k = J^T(x_{k+1}) f(x_{k+1}) - J^T(x_k) f(x_k). \quad (3.10)$$

A condição secante (3.8) segue directamente a partir da condição secante usual, (2.20). A segunda, (3.9), é proposta por Dennis, Gay e Welsch [8, Capítulo 10], e baseia-se na estrutura especial da segunda parte da matriz Hessiana. Esta condição secante surgiu da seguinte motivação. Idealmente, A_{k+1} deve ser uma boa aproximação ao termo exacto de segunda ordem no ponto $x = x_{k+1}$, que é,

$$A_{k+1} \approx \sum_{j=1}^m f_j(x_{k+1}) \nabla^2 f_j(x_{k+1}).$$

Uma vez que não pretendemos calcular as Hessianas individuais $\nabla^2 f_j$ nesta fórmula, substitui-se cada uma delas por uma aproximação $(B_j)_{k+1}$ e impõem-se a condição que $(B_j)_{k+1}$ deve simular o comportamento da sua parte correspondente $\nabla^2 f_j$ sobre os passos acabados de realizar; que é,

$$\begin{aligned} (B_j)_{k+1} (x_{k+1} - x_k) &= \nabla f_j(x_{k+1}) - \nabla f_j(x_k) \\ &= (\text{coluna } j \text{ de } J(x_{k+1})^T) - (\text{coluna } j \text{ de } J(x_k)^T). \end{aligned}$$

Esta condição conduz-nos a uma condição secante para A_{k+1} , nomeadamente,

$$\begin{aligned} A_{k+1} (x_{k+1} - x_k) &= \sum_{j=1}^m f_j(x_{k+1}) (B_j)_{k+1} (x_{k+1} - x_k) \\ &= \sum_{j=1}^m f_j(x_{k+1}) (\text{coluna } j \text{ de } J(x_{k+1})^T) - (\text{coluna } j \text{ de } J(x_k)^T) \\ &= (J(x_{k+1}) - J(x_k))^T f(x_{k+1}). \end{aligned}$$

Tal como nos métodos quasi-Newton, a condição secante por si só não é suficiente para determinar A_{k+1} de modo único. Tem que ser impostas condições adicionais para especificar um determinado método quasi-Newton estruturado. Por exemplo, Dennis, Gay e Welsch impõem a condição de que A_{k+1} deve ser simétrica e que, entre todas as matrizes A que satisfazem a condição secante (3.9) e simetria, A_{k+1} é, a que se encontra mais próxima da matriz actual A_k .

A primeira fórmula de actualização foi proposta por Broyden e Dennis (BD) em 1973 [11], e segue directamente a partir da condição secante (3.8). A segunda foi proposta

por Bartholomew-Biggs em 1977 [11], e a terceira foi proposta por Dennis, Gay e Welsch (DGW) em 1981 [11], e baseiam-se na condição secante (3.9).

As fórmulas de actualização quasi-Newton estruturadas são, geralmente, de característica-um ou característica-dois.

Fórmula de actualização BD:

$$A_{k+1} = A_k + \frac{(u_k - A_k s_k) s_k^T + s_k (u_k - A_k s_k)^T}{s_k^T s_k} - \frac{(u_k - A_k s_k)^T s_k}{(s_k^T s_k)^2} s_k s_k^T.$$

Recentemente (anos 80), e usando técnicas de escalonamento, Bartholomew-Biggs e Dennis, Gay e Welsch propuseram algoritmos robustos que resolvem tanto problemas de grandes resíduos como de pequenos resíduos. Quando os resíduos são grandes, os seus algoritmos desempenham tão bem como o método de Broyden e Dennis. Por outro lado, para os problemas de resíduos muito pequenos, os seus algoritmos desempenham quase tão bem como o do método de Gauss-Newton. As suas fórmulas de actualização são as seguintes.

Fórmula de actualização de Biggs:

$$\begin{aligned} A_{k+1} &= \beta_k A_k + \frac{(v_k - \beta_k A_k s_k) (v_k - \beta_k A_k s_k)^T}{(v_k - \beta_k A_k s_k)^T s_k} \\ \beta_k &= \frac{f^T(x_{k+1}) f(x_k)}{f^T(x_k) f(x_k)} \end{aligned} \quad (3.11)$$

Fórmula de actualização DGW:

$$\begin{aligned} A_{k+1} &= \beta_k A_k + \frac{(v_k - \beta_k A_k s_k) y_k^T + y_k (v_k - \beta_k A_k s_k)^T}{y_k^T s_k} - \frac{s_k^T (v_k - \beta_k A_k s_k)}{(y_k^T s_k)^2} y_k y_k^T \\ \beta_k &= \min \left(\left| \frac{s_k^T v_k}{s_k^T A_k s_k} \right|, 1 \right) \end{aligned} \quad (3.12)$$

onde β_k é um factor de escalonamento.

Algoritmo quasi-Newton estruturado: DGW

1. Dar x_1 e dar uma matriz inicial de aproximação A_1 (por ex. $A_1=I$). Calcular $J(x_1)$, $f(x_1)$, $\nabla F(x_1)=J(x_1)^T f(x_1)$. Fazer $k = 1$

2. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares

$$(J^T(x_k) J(x_k) + A_k) d_k = -\nabla F(x_k)$$

3. Calcular o comprimento do passo, α_k , através de um procedimento de procura unidimensional
4. Calcular $x_{k+1} = x_k + \alpha_k d_k$
5. Se x_{k+1} satisfaz o critério de paragem, terminar $[x^* \leftarrow x_{k+1} \text{ e } F^* \leftarrow F(x_{k+1})]$; senão ir para 6.
6. Calcular $J(x_{k+1})$, $f(x_{k+1})$, $\nabla F(x_{k+1}) = J(x_{k+1})^T f(x_{k+1})$, $s_k = x_{k+1} - x_k$, $y_k = \nabla F(x_{k+1}) - \nabla F(x_k)$, $v_k = (J(x_{k+1}) - J(x_k))^T f(x_{k+1})$
7. Actualizar A_{k+1} usando a fórmula de actualização DGW; fazer $k = k + 1$ e voltar para 2

3.2.4 Métodos quasi-Newton estruturados e factorizados

Algumas actualizações quasi-Newton, propostas para optimização não linear geral, tentam conservar as matrizes definidas positivas, como por exemplo, as fórmulas de actualização BFGS e DFP. Esta propriedade garante que a direcção de procura é uma direcção descendente para F . Por outro lado, nas actualizações quasi-Newton estruturadas, não é fácil construir uma actualização para A_k de modo que a matriz $J(x_k)^T J(x_k) + A_k$ seja definida positiva. Com o objectivo de conservar a matriz do sistema (3.7) definida positiva, têm sido propostas várias estratégias, como a decomposição modificada de Cholesky da matriz $J(x_k)^T J(x_k) + A_k$.

No trabalho de Yabe e Takahashi [11] e posteriormente em [10] é proposta uma aproximação directa no sentido de manter a matriz dos coeficientes (3.7) definida positiva. Propõe-se o cálculo da direcção de procura, d_k , através da resolução do seguinte sistema de equações lineares

$$(J(x_k) + L_k)^T (J(x_k) + L_k) d_k = -J(x_k)^T f(x_k),$$

em que L_k é uma matriz $m \times n$ por forma a que $L_k^T L_k + L_k^T J(x_k) + J(x_k)^T L_k$ seja uma aproximação à segunda parte da matriz Hessiana de F , na iteração k . Como a matriz dos coeficientes do sistema surge na forma factorizada, espera-se que a direcção de procura seja descendente para a função F .

A matriz L_k é actualizada de modo que a nova matriz L_{k+1} satisfaz a **condição secante**,

$$(J(x_{k+1}) + L_{k+1})^T (J(x_{k+1}) + L_{k+1}) s_k = z_k \quad (3.13)$$

onde

$$z_k = y_k \quad (3.14)$$

ou

$$z_k = v_k + J(x_{k+1})^T J(x_{k+1}) s_k, \quad (3.15)$$

e os vectores s_k , y_k , v_k são definidos por,

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= J(x_{k+1})^T f(x_{k+1}) - J(x_k)^T f(x_k) \\ v_k &= (J(x_{k+1}) - J(x_k))^T f_{k+1}. \end{aligned} \quad (3.16)$$

Se z_k é dado por (3.14), a condição secante segue directamente a partir da condição secante usual, (2.20), para os métodos quasi-Newton para optimização não linear geral. Se z_k é dado por (3.15), a condição secante segue directamente a partir da condição secante proposta por Dennis, Gay e Welsch, (3.9), para os métodos quasi-Newton estruturados.

É possível mostrar que, para s_k e z_k diferentes de zero, a equação matricial (3.13) é consistente apenas se

$$L_{k+1}^T h = z_k - J(x_{k+1})^T h \quad \text{e} \quad L_{k+1} s_k = h - J(x_{k+1}) s_k \quad (3.17)$$

para algum vector h de dimensão m . Estas equações têm uma matriz solução comum L_{k+1} se e só se cada equação separadamente tem uma solução e $h^T h = s_k^T z_k$. Assim, o objectivo é encontrar uma matriz L_{k+1} que satisfaça as equações (3.17) pressupondo que $s_k^T z_k > 0$. Uma vez que (3.17) não tem uma solução L_{k+1} única, usa-se a técnica da variação mínima, para determinar a correspondente fórmula de actualização. A técnica da variação mínima consiste em minimizar a variação verificada na matriz L , da iteração k para a iteração $k+1$, sujeita a um conjunto de restrições que caracterizam a fórmula actualizadora. Estas restrições incluem não só a condição secante como podem, por exemplo, garantir que a matriz se conserve simétrica ou mesmo definida positiva [5].

Fórmula de actualização factorizada do tipo L (BFGS):

Para qualquer vector h , desconhecido de dimensão m , que verifique $h^T h = s_k^T z_k$, o mínimo relativamente a L_{k+1} da norma de Frobenius da variação verificada na matriz L_k , sujeito à condição secante (3.13), é formulado por

$$\min_{L_{k+1}} \|L_{k+1} - L_k\|_F$$

sujeito a $L_{k+1}^T h = z_k - J(x_{k+1})^T h$. Yabe e Takahashi obtiveram então a seguinte fórmula de actualização, de característica um, para L_k :

$$L_{k+1} = L_k + \frac{(L_k + J(x_{k+1})) s_k}{b_k} \left(\sqrt{\frac{b_k}{a_k}} z_k - B_k^\# s_k \right)^T \quad (3.18)$$

onde $B_k^\# = (J(x_{k+1}) + L_k)^T (J(x_{k+1}) + L_k)$, e a_k e b_k são escalares definidos por $a_k = s_k^T z_k$ e $b_k = s_k^T B_k^\# s_k$.

Fazendo,

$$B_{k+1} = (J(x_{k+1}) + L_{k+1})^T (J(x_{k+1}) + L_{k+1}),$$

tem-se

$$B_{k+1} = B_k^\# - \frac{B_k^\# s_k s_k^T B_k^\#}{b_k} + \frac{z_k z_k^T}{a_k}, \quad (3.19)$$

a qual tem um aspecto semelhante à fórmula de actualização BFGS para optimização não linear geral. Na fórmula de actualização (3.19), para problemas de mínimos quadrados não lineares, $B_k^\#$ aparece no lugar de B_k da fórmula BFGS original. Por esta razão é designada por fórmula factorizada do tipo-BFGS. A diferença essencial entre a fórmula BFGS original e (3.19) é que a estrutura da matriz Hessiana (3.5) está incluída em z_k e $B_k^\#$.

Fórmula de actualização factorizada do tipo L (DFP):

Introduzindo duas matrizes não singulares $W_E \in \mathbb{R}^{m \times m}$ (ou $\in \mathbb{R}^{n \times n}$), $W_D \in \mathbb{R}^{n \times n}$ na formulação da função objectivo de variação mínima, sujeito à condição secante (3.13), Yabe e Takahashi obtiveram uma fórmula de actualização factorizada do tipo-DFP, que designaram por tipo L (DFP).

Assim, para qualquer vector h , desconhecido de dimensão m que verifique $h^T h = s_k^T z_k$, e para as matrizes não singulares W_E e W_D , minimizando

$$\|W_E (L_{k+1} - L_k) W_D\|_F$$

relativamente a L_{k+1} , sujeito à condição $L_{k+1} s_k = h - J(x_{k+1}) s_k$, obtiveram a seguinte fórmula de actualização, de característica um, para L_k :

$$L_{k+1} = L_k + (L_k + J(x_{k+1})) \left[\sqrt{\frac{a_k}{c_k}} \left(B_k^\# \right)^{-1} z_k - s_k \right] \frac{z_k^T}{a_k} \quad (3.20)$$

onde a_k e c_k são escalares definidos por $a_k = s_k^T z_k$ e $c_k = z_k^T \left(B_k^\# \right)^{-1} z_k$.

Fazendo,

$$B_{k+1} = (J(x_{k+1}) + L_{k+1})^T (J(x_{k+1}) + L_{k+1}),$$

tem-se

$$B_{k+1} = B_k^\# + \left(1 + \frac{b_k}{a_k}\right) \frac{z_k z_k^T}{a_k} - \frac{1}{a_k} \left(B_k^\# s_k z_k^T + z_k s_k^T B_k^\#\right), \quad (3.21)$$

a qual tem um aspecto semelhante à fórmula de actualização DFP para optimização não linear geral. Na fórmula de actualização (3.21), para problemas de mínimos quadrados não lineares, $B_k^\#$ aparece no lugar de B_k da fórmula DFP original. A diferença essencial entre a fórmula DFP original e (3.21) é que a estrutura da matriz Hessiana (3.5) está incluída em z_k e $B_k^\#$.

A análise de convergência do método quasi-Newton factorizado baseado nas fórmulas apresentadas estabelece:

- convergência q-superlinear local \rightarrow quer os problemas sejam: de resíduos nulos, de pequenos resíduos, fortemente não lineares ou de grandes resíduos.

Factor de escalonamento da matriz

Para problemas de resíduos nulos, as matrizes A_k e neste caso $L_k^T L_k + L_k^T J(x_k) + J(x_k)^T L_k$ devem convergir para zero à medida que o processo converge para a solução de (3.1), uma vez que elas aproximam a segunda parte da Hessiana de F . Se os elementos dessas matrizes não se tornam, ao longo do processo iterativo, cada vez mais pequenos, então os métodos quasi-Newton estruturados e os factorizados não conseguem competir com o método de Gauss-Newton.

Como as actualizações do tipo L (BFGS) e do tipo L (DFP) nunca geram a matriz nula, torna-se necessário introduzir algum esquema que force essa convergência. Os esquemas de escalonamento de matrizes actualizadoras mais conhecidos foram introduzidos por Bartholomew-Biggs, em 1977, (ver [1]) e Dennis, Gay e Welsch, em 1981, ([3]) (DGW). O factor de escalonamento de Biggs é dado por,

$$\beta_k = \frac{f^T(x_{k+1})f(x_k)}{f^T(x_k)f(x_k)} \quad (3.22)$$

e baseia-se na ideia de que se $f(x_{k+1}) = \beta_k f(x_k)$, para algum β_k , se $A_k \approx \sum_{j=1}^m f_j(x_k) \nabla^2 f_j(x_k)$ e se cada f_j é uma forma quadrática, então $\sum_{j=1}^m f_j(x_{k+1}) \nabla^2 f_j(x_{k+1}) \approx \beta_k A_k$. Para as fórmulas de actualização do tipo L (BFGS/DFP) apresentadas e usando o mesmo argumento, β_k toma também a forma (3.22).

A estratégia de Dennis, Gay e Welsch pressupõe que o espectro da matriz escalonada $\beta_k A_k$ é idêntico ao da segunda parte da matriz Hessiana na direcção de s_k . Assim, os autores obtiveram o factor (3.12) usando a relação

$$\begin{aligned}
& \left| \left[s_k^T \left\{ \sum_{j=1}^m f_j(x_{k+1}) \nabla^2 f_j(x_{k+1}) \right\} s_k / s_k^T s_k \right] \left[s_k^T (\beta_k A_k) s_k / s_k^T s_k \right]^{-1} \right| \\
&= |s_k^T v_k / s_k^T (\beta_k A_k) s_k| = 1,
\end{aligned}$$

onde o vector v_k é dado em (3.9). Os métodos quasi-Newton estruturados com os factores de escalonamento (3.11) e (3.12) são razoáveis no sentido de que se a função $f(x_{k+1})$ é zero, então $v_k = 0$ e $\beta_k = 0$, portanto a nova matriz A_{k+1} também é zero. Este facto é baseado no uso da condição secante (3.9).

Dado que o factor de escalonamento DGW, (3.12), contém a matriz A_k , não é possível usar directamente esta fórmula nas versões factorizadas apresentadas. Contudo, nas versões factorizadas do tipo L (BFGS/DFP), pode considerar-se uma estratégia similar à de DGW. Assim, o factor β_k deve ser escolhido por forma a que a matriz

$$(\beta_k L_k)^T (\beta_k L_k) + (\beta_k L_k)^T J(x_{k+1}) + J(x_{k+1})^T (\beta_k L_k)$$

tenha um espectro idêntico ao da segunda parte da matriz Hessiana de F , na direcção de s_k . Assim, tem-se a seguinte relação,

$$\frac{|s_k^T v_k|}{s_k^T \left[(\beta_k L_k)^T (\beta_k L_k) + (\beta_k L_k)^T J(x_{k+1}) + J(x_{k+1})^T (\beta_k L_k) \right] s_k} = 1,$$

que origina

$$\beta_k = \frac{-(L_k s_k)^T J(x_{k+1}) s_k + \text{sign} \left((L_k s_k)^T J(x_{k+1}) s_k \right) \sqrt{\phi_k}}{\|L_k s_k\|_2^2},$$

onde

$$\phi_k = \left((L_k s_k)^T J(x_{k+1}) s_k \right)^2 + \|L_k s_k\|_2^2 |s_k^T v_k|$$

e o simbolo $\text{sign}(\zeta)$ representa o sinal de ζ . Na prática, é mais vantajoso implementar o factor de escalonamento,

$$\beta_k = \min \left(\frac{\left| -(L_k s_k)^T J(x_{k+1}) s_k + \text{sign} \left((L_k s_k)^T J(x_{k+1}) s_k \right) \sqrt{\phi_k} \right|}{\|L_k s_k\|_2^2}, 1 \right). \quad (3.23)$$

Se bem que esta ideia seja interessante, o custo computacional associado é muito maior do que no factor de Biggs (3.22).

Introduzindo o factor (3.22) ou (3.23) nas fórmulas de actualização do tipo L (BFGS) (veja-se em (3.18)) e do tipo L (DFP) (veja-se em (3.20)) tem-se, respectivamente

$$L_{k+1} = \beta_k L_k + \frac{(\beta_k L_k + J(x_{k+1})) s_k}{b_k} \left(\sqrt{\frac{b_k}{a_k}} z_k - B_k^\# s_k \right)^T \quad (3.24)$$

$$L_{k+1} = \beta_k L_k + (\beta_k L_k + J(x_{k+1})) \left[\sqrt{\frac{a_k}{c_k}} \left(B_k^\# \right)^{-1} z_k - s_k \right] \frac{z_k^T}{a_k} \quad (3.25)$$

onde a_k , b_k e c_k são os escalares definidos por $a_k = s_k^T z_k$, $b_k = s_k^T B_k^\# s_k$, $c_k = z_k^T \left(B_k^\# \right)^{-1} z_k$, z_k é dado por (3.15) e a matriz $B_k^\#$ é reescrita como

$$B_k^\# = (J(x_{k+1}) + \beta_k L_k)^T (J(x_{k+1}) + \beta_k L_k).$$

Algoritmo quasi-Newton estruturado e factorizado

Apresenta-se de seguida o algoritmo que corresponde ao método quasi-Newton estruturado e factorizado baseado nas actualizações factorizadas do tipo L (BFGS), (3.18) ou (3.24), e tipo L (DFP), em (3.20) ou (3.25).

Algoritmo QNEF

1. Iniciar com o ponto $x_1 \in \mathbb{R}^n$ e uma matriz L_1 $m \times n$. Fazer $k = 1$
2. Determinar a direcção de procura, d_k , resolvendo o sistema de equações lineares, $(J(x_k) + L_k)^T (J(x_k) + L_k) d_k = -J(x_k)^T f(x_k)$
3. Escolher o comprimento do passo, α_k , através de um algoritmo de procura unidimensional
4. Fazer $x_{k+1} = x_k + \alpha_k d_k$
5. Se o novo ponto satisfaz o critério de paragem então parar [$x^* \leftarrow x_{k+1}$ e $F^* \leftarrow F(x_{k+1})$]; senão ir para 6.
6. Actualizar L_{k+1} usando a correspondente fórmula de actualização; fazer $k = k + 1$ e voltar para 2

3.2.5 Critério de paragem

Em [11] é proposto o seguinte critério de paragem: o processo iterativo termina

- se $\|f(x_{k+1})\|_{\infty} \leq \max\{\varepsilon_1, \text{'zero'}\}$; ou
- se $\left| e_j^T J(x_{k+1})^T f(x_{k+1}) \right| \leq \max\{\varepsilon_2, \text{'zero'}\} \|f(x_{k+1})\| \|J(x_{k+1}) e_j\|$ para $j=1, 2, \dots, n$ e $\|x_{k+1} - x_k\|_{\infty} \leq \max\{\varepsilon_3, \text{'zero'}\} \max\{\|x_{k+1}\|_{\infty}, 1\}$, onde e_j representa a j -ésima coluna da matriz identidade; ou
- se o número de iterações excede o limite máximo pré-estabelecido (itMáx),

onde $\|\cdot\|_{\infty}$ representa a norma infinita e 'zero' denota a precisão da máquina.

3.3 Exercícios

1. Implementar o método de Gauss-Newton, com o objectivo de ajustar o melhor possível o modelo matemático

$$M(i; x_1, x_2) = x_1 + x_1 x_2 i$$

aos dados apresentados na tabela

i	-1	0	1
y_i	1	2	3

no sentido dos mínimos quadrados. Como aproximações iniciais aos parâmetros use $(1.5, 0.75)^T$. Páre o processo iterativo quando o critério de paragem for verificado para $\varepsilon_1 = \varepsilon_2 = 0.5$. Deve, implementar o algoritmo de procura unidimensional baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$. Nos casos em que a matriz $J_k^T J_k$ não é definida positiva, usar a factorização modificada de Cholesky desta matriz.

2. Implementar duas iterações do método de Gauss-Newton, com o objectivo de ajustar o melhor possível o modelo matemático

$$M(i; x_1, x_2) = x_1(1 - x_2^i)$$

aos dados apresentados na tabela

i	1	2	3
y_i	1.5	2.25	2.625

no sentido dos mínimos quadrados. Como aproximações iniciais aos parâmetros use $(2, 0.1)^T$. Deve, implementar o algoritmo de procura unidimensional baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$. Nos casos em que a matriz $J_k^T J_k$ não é definida positiva, usar a factorização modificada de Cholesky desta matriz. No critério de paragem usar $\varepsilon_1 = \varepsilon_2 = 10^{-4}$.

3. Implementar duas iterações do método quasi-Newton estruturado baseado na fórmula de actualização de Biggs, com o objectivo de ajustar o melhor possível o modelo matemático

$$M(i; x_1, x_2) = x_1(1 - x_2^i)$$

aos dados apresentados na tabela

i	1	2	3
y_i	1.5	2.25	2.625

no sentido dos mínimos quadrados. Como aproximações iniciais aos parâmetros use $(2, 0.1)^T$. Considere o factor de escalonamento de Biggs dado por

$$\beta_k = \frac{|f^T(x_{k+1})f(x_k)|}{\|f(x_k)\|_2^2}$$

e $A_1 = 0$. Deve, implementar o algoritmo de procura unidimensional baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$. Nos casos em que a matriz $(J_k^T J_k + A_k)$ não é definida positiva, usar a factorização modificada de Cholesky desta matriz. No critério de paragem usar $\varepsilon_1 = \varepsilon_2 = 10^{-4}$.

4. Implementar duas iterações do método quasi-Newton estruturado baseado na fórmula de actualização de DGW, com o objectivo de ajustar o melhor possível o modelo matemático

$$M(i; x_1, x_2) = x_1(1 - x_2^i)$$

aos dados apresentados na tabela

i	1	2	3
y_i	1.5	2.25	2.625

no sentido dos mínimos quadrados. Como aproximações iniciais aos parâmetros use $(2, 0.1)^T$ e $A_1 = 0$. Deve, implementar o algoritmo de procura unidimensional baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$. Nos casos em que a matriz $(J_k^T J_k + A_k)$ não é definida positiva, usar a factorização modificada de Cholesky desta matriz. No critério de paragem usar $\varepsilon_1 = \varepsilon_2 = 10^{-4}$.

5. Implementar duas iterações do método quasi-Newton estruturado e factorizado baseado na fórmula de actualização tipo L (BFGS), dada por (3.24) com o vector z_k definido por (3.14), com o objectivo de ajustar o melhor possível o modelo matemático

$$M(i; x_1, x_2) = x_1 e^{x_2 t_i}$$

aos dados apresentados na tabela

t_i	1	2	4	5	8
y_i	3.2939	4.2699	7.1749	9.3008	20.259

no sentido dos mínimos quadrados. Como aproximações iniciais aos parâmetros use $(2.05, 0.25)^T$ e $L_1 = 0$. Deve, implementar o algoritmo de procura unidimensional baseado no critério de Armijo para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.1$. No critério de paragem usar $\varepsilon_1 = \varepsilon_2 = 10^{-4}$.

6. Escreva um programa em MatLab para resolver um problema de mínimos quadrados não lineares usando o algoritmo quasi-Newton estruturado e factorizado baseado na fórmula de actualização do tipo L(BFGS). Use $L_0 = 0$ como aproximação inicial ao segundo termo da Hessiana. Inclua o seguinte:

- Teste o algoritmo com os dois vectores z_k : (3.14) e (3.15).
- Teste o algoritmo sem e com factor de escalonamento (use o factor (3.23)).
- Use a estratégia de redução sucessiva do passo apresentada na subsecção 2.3.1 com $\mu_1 = 0.1$.
- Utilize a estratégia de *skipping* apresentada no capítulo 2 com parâmetro de recomeço $\gamma = 10^{-12}$.
- Aceite x como uma solução do problema de optimização se o critério de paragem descrito na secção anterior se verifica. Use $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-4}$ e itMax=200.
- Escreva o valor inicial, e em cada iteração escreva a direcção de procura, o comprimento do passo α e a nova aproximação à solução x_{k+1} . Indique se não foi encontrada nenhuma solução ao fim do limite máximo de iterações pré-estabelecido.
- Teste os algoritmos com os seguintes problemas teste:

- Função de Rosenbrock ($n = 2, m = 2$)
 - $F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
 - $x_0 = (-1.2, 1)^T$.

- Função de Beale ($n = 2, m = 3$)
 - $F(x) = \sum_{i=1}^3 f_i^2(x)$ com $f_i(x) = y_i - x_1(1 - x_2^i)$
onde $y_1 = 1.5, y_2 = 2.25, y_3 = 2.625$
 - $x_0 = (0.1, 0.1)^T$

- Função de Bard ($n = 3, m = 15$)
 - $F(x) = \sum_{i=1}^{15} f_i^2(x)$ com $f_i(x) = y_i - \left(x_1 + \frac{u_i}{v_i x_2 + w_i x_3} \right)$
onde $u_i = i, v_i = 16 - i, w_i = \min(u_i, v_i)$, e

i	y_i	i	y_i	i	y_i
1	0.14	6	0.32	11	0.73
2	0.18	7	0.35	12	0.96
3	0.22	8	0.39	13	1.34
4	0.25	9	0.37	14	2.10
5	0.29	10	0.58	15	4.39

- $x_0 = (1, 1, 1)^T$

- Função de Jennrich ($n = 2, m = 10$)

- $F(x) = \sum_{i=1}^{10} f_i^2(x)$ com $f_i(x) = y_i - (\exp(ix_1) + \exp(ix_2))$

- onde $y_i = 2 + 2i$

- $x_0 = (0.3, 0.4)^T$

- Função de Osborne1 ($n = 5, m = 33$)

$$- F(x) = \sum_{i=1}^{15} f_i^2(x) \quad \text{com} \quad f_i(x) = y_i - (x_1 + x_2 \exp(-t_i x_4) + x_3 \exp(-t_i x_5))$$

onde $t_i = 10(i-1)$ e

i	y_i	i	y_i	i	y_i	i	y_i
1	0.844	10	0.784	19	0.538	28	0.431
2	0.908	11	0.751	20	0.522	29	0.424
3	0.932	12	0.718	21	0.506	30	0.420
4	0.936	13	0.685	22	0.490	31	0.414
5	0.925	14	0.658	23	0.478	32	0.411
6	0.908	15	0.628	24	0.467	33	0.406
7	0.881	16	0.603	25	0.457		
8	0.850	17	0.580	26	0.448		
9	0.818	18	0.558	27	0.438		

$$- x_0 = (0.5, 1.5, -1, 0.01, 0.02)^T$$

- Função Gaussiana ($n = 3, m = 15$)

$$- F(x) = \sum_{i=1}^{15} f_i^2(x) \quad \text{com} \quad f_i(x) = x_1 \exp\left(\frac{-x_2 (t_i - x_3)^2}{2}\right) - y_i$$

onde $t_i = (8-i)/2$ e

i	y_i
1, 15	0.0009
2, 14	0.0044
3, 13	0.0175
4, 12	0.0540
5, 11	0.1295
6, 10	0.2420
7, 9	0.3521
8	0.3989

$$- x_0 = (0.4, 1, 0)^T$$

Capítulo 4

Optimização não linear com restrições

4.1 Forma geral do problema

A formulação matemática de um problema de optimização não linear com restrições, na sua forma mais geral, é

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimizar}} && F(x) \\ \text{sujeito a} &&& h_i(x) = 0, \quad i = 1, \dots, m \\ &&& h_i(x) \geq 0, \quad i = 1, \dots, \overline{m} \end{aligned} \tag{4.1}$$

onde função objectivo $F : \mathbb{R}^n \rightarrow \mathbb{R}$ e as funções de restrição $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ assumimos que são duas vezes continuamente diferenciáveis num subconjunto de \mathbb{R}^n . As funções h_i , com $i = 1, \dots, m$, integram as *restrições de igualdade* e as h_i , com $i = 1, \dots, \overline{m}$, integram as *restrições de desigualdade*. Qualquer conjunto de equações e inequações pode ser escrito na forma (4.1). Em particular, qualquer inequação do tipo “menor ou igual” pode ser transformada numa restrição equivalente do tipo “maior ou igual”. Tais transformações são meramente ‘cosméticas’, mas simplificam a notação para descrever as restrições.

Exemplo 4.1 *Para o seguinte problema*

$$\begin{aligned} & \underset{x \in \mathbb{R}^5}{\text{minimizar}} && e^{(x_1 x_2 x_3 x_4 x_5)} \\ \text{s.a} &&& x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10 \\ &&& x_2 x_3 = 5 x_4 x_5 \\ &&& x_2 x_3 = 5 x_4 x_5 \end{aligned}$$

tem-se $n = 5$; $i = 1, 2, 3$ e

$$\begin{aligned} F(x) &= e^{(x_1 x_2 x_3 x_4 x_5)} \\ h_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 \\ h_2(x) &= x_2 x_3 - 5 x_4 x_5 \\ h_3(x) &= x_2 x_3 - 5 x_4 x_5. \end{aligned}$$

4.2 Notação

As derivadas das funções envolvidas no problema (4.1) definem os seguintes vectores e matrizes.

O vector gradiente de F , ∇F , definido em (1.2); a matriz Hessiana de F , $\nabla^2 F$, definida em (1.3); os vectores dos gradientes das restrições h_i , ∇h_i para $i = 1, \dots, m + \overline{m}$, dados por

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

a matriz que contém os gradientes das restrições, ao longo das linhas, dada por

$$\nabla h(x) = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

e as matrizes Hessianas das restrições h_i , $\nabla^2 h_i$ para $i = 1, \dots, m + \overline{m}$, dados por

$$\nabla^2 h_i(x) = \begin{pmatrix} \frac{\partial^2 h_i}{\partial x_1^2} & \frac{\partial^2 h_i}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 h_i}{\partial x_1 \partial x_n} \\ \frac{\partial^2 h_i}{\partial x_2 \partial x_1} & \frac{\partial^2 h_i}{\partial x_2^2} & \dots & \frac{\partial^2 h_i}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 h_i}{\partial x_n \partial x_1} & \frac{\partial^2 h_i}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 h_i}{\partial x_n^2} \end{pmatrix}.$$

4.3 Condições de Optimalidade

Nesta secção, enunciam-se as condições satisfeitas pelas soluções do problema de optimização com restrições. Pelas razões já apontadas no caso sem restrições, apenas dizem respeito a soluções locais. No caso de problemas convexos, que é, quando a região admissível é convexa e F é uma função convexa, qualquer solução local é também uma solução global.

Começa-se por introduzir alguma terminologia referida ao longo desta secção.

- Um ponto que satisfaz todas as restrições diz-se *ponto admissível*.

- Define-se *conjunto admissível* ou *região admissível*, \mathcal{D} , como sendo o conjunto de todos os pontos admissíveis x , isto é,

$$\mathcal{D} = \{x : h_i(x) = 0, \ i = 1, \dots, m; \ h_i(x) \geq 0, \ i \in \{1, \dots, \overline{m}\}\}.$$

- Num ponto admissível \bar{x} , uma restrição de desigualdade $h_i(x) \geq 0$, com $i \in \{1, \dots, \overline{m}\}$, diz-se:
 - *activa* se $h_i(\bar{x}) = 0$ e
 - *não activa* se $h_i(\bar{x}) > 0$.

No primeiro caso, o ponto \bar{x} diz-se sobre a fronteira da *restrição*, e no último no *interior* da restrição. Em particular, todas as restrições de igualdade são activas em qualquer ponto admissível. O conjunto de pontos admissíveis para o qual pelo menos uma restrição está activa diz-se *fronteira* da região admissível. Todos os outros pontos são *pontos interiores*. Os pontos interiores são apenas “interiores” para as restrições de desigualdade, uma vez que não é possível ser interior a uma restrição de igualdade. Assim, alguns autores preferem a designação de *pontos interiores relativos*.

- Define-se *conjunto activo*, $\mathcal{A}(x)$, em qualquer ponto admissível x , como sendo a união do conjunto de índices $\{1, \dots, m\}$ com os índices das restrições de desigualdade activas, isto é,

$$\mathcal{A}(x) = \{1, \dots, m\} \cup \{i \in \{1, \dots, \overline{m}\} : h_i(x) = 0\}. \quad (4.2)$$

- O problema (4.1) pode ser reescrito mais compactamente na forma

$$\text{minimizar}_{x \in \mathcal{D}} F(x). \quad (4.3)$$

Considerem-se as seguintes definições:

Definição 4.1 Um vector x^* é uma solução local do problema (4.3) se $x^* \in \mathcal{D}$ e existe uma vizinhança \mathbb{V} de x^* tal que $F(x^*) \leq F(x)$ para $x \in \mathbb{V} \cap \mathcal{D}$.

Definição 4.2 Um vector x^* é uma solução local estrita (também designada uma solução local forte) se $x^* \in \mathcal{D}$ e existe uma vizinhança \mathbb{V} de x^* tal que $F(x^*) < F(x)$ para todo $x \in \mathbb{V} \cap \mathcal{D}$ com $x \neq x^*$.

Definição 4.3 Um vector x^* é uma solução local isolada se $x^* \in \mathcal{D}$ e existe uma vizinhança \mathbb{V} de x^* tal que x^* é o único minimizante em $\mathbb{V} \cap \mathcal{D}$.

No problema sem restrições, as condições de optimalidade são deduzidas usando uma aproximação em série de Taylor para analisar o comportamento da função objectivo F em torno do minimizante local x^* . Em particular, em pontos “próximos” de x^* o valor de F não decresce.

Em problemas com restrições, a estratégia é idêntica. Usam aproximações em série de Taylor para analisar o comportamento da função objectivo F e das restrições h_i em torno do minimizante local x^* . Neste caso, em pontos *admissíveis* “próximos” de x^* o valor de F não decresce.

As condições de optimalidade são apresentadas genericamente para quaisquer restrições lineares/não lineares do tipo igualdade/desigualdade envolvidas no problema.

Interessa, assim, analisar o comportamento da função objectivo quando a partir de um ponto admissível x^* , se move em direcção a outros pontos admissíveis próximos.

É importante referir que se todas as restrições fossem lineares, os movimentos admissíveis seriam completamente caracterizados por direcções admissíveis. Em termos matemáticos, p é uma direcção admissível se existe algum $\varepsilon > 0$ tal que $x^* + ap \in \mathcal{D}$ para todo $0 \leq a \leq \varepsilon$. Portanto, um pequeno movimento a partir de x^* ao longo de uma direcção admissível mantém a admissibilidade. Num minimizante local não podem existir direcções admissíveis descentes para F , por isso

$$p^T \nabla F(x) \geq 0, \quad \text{para todas as direcções admissíveis } p \text{ em } x^*.$$

Se o problema tem restrições não lineares, pode não ser mais possível um movimento para pontos próximos de x^* ao longo de direcções admissíveis. Em vez disso, pequenos movimentos admissíveis serão feitos ao longo de *curvas admissíveis*. Uma *curva* é um conjunto de pontos $\{x = x(t) : t_0 \leq t \leq t_1\}$. Uma *curva* diz-se *admissível* relativamente ao conjunto de restrições se $h_i(x(t)) = 0$, para $i = 1, \dots, m$ e $h_i(x(t)) \geq 0$, para $i = 1, \dots, \bar{m}$, para todo $t_0 \leq t \leq t_1$. Uma curva que passa em x^* satisfaz $x(t^*) = x^*$ para algum $t_0 \leq t \leq t_1$. Fazendo um deslocamento do parâmetro t , pode assumir-se sempre que $t^* = 0$.

No caso de restrições de desigualdade o movimento é feito ao longo de um *arco admissível* pois, muitas vezes, o movimento só é possível apenas num sentido. Formalmente, define-se um arco que começa em (parte de) x^* como uma curva dirigida $x(t)$ parametrizada pela variável t num intervalo $[0, T]$ na qual $x(0) = x^*$. Mesmo assim, a ideia básica é que o valor da função objectivo não deverá diminuir em pontos admissíveis próximos de x^* .

4.3.1 Condições de optimalidade de 1ª ordem

As condições de optimalidade serão dadas em termos da função Lagrangeana. A função Lagrangeana é uma única função que combina a função objectivo e as funções de restrição. Para o problema com restrições (4.1) a função Lagrangeana é definida por

$$\mathbb{L}(x, \lambda) = F(x) - \sum_{i=1}^{m+\overline{m}} \lambda_i h_i(x) \quad (4.4)$$

onde λ é o vector dos multiplicadores de Lagrange e $\{h_i : i = 1, \dots, m + \overline{m}\}$ são as funções de restrição.

É necessário analisar as propriedades dos gradientes das restrições. O vector $\nabla h_i(x)$ é muitas vezes designado por *normal* à restrição h_i no ponto x , pois é, em geral, um vector perpendicular aos contornos da restrição h_i em x , e no caso de uma restrição de desigualdade, o vector aponta no sentido do lado admissível da restrição. É possível, contudo, que $\nabla h_i(x)$ possa ser nulo devido à representação algébrica de h_i , de modo que o termo $\lambda_i \nabla h_i(x)$ é nulo para todos os valores de λ_i e consequentemente não surgirá no gradiente da Lagrangeana $\nabla_x \mathbb{L}$. Para evitar este comportamento degenerado no valor de x em questão, geralmente, assume-se que x é ponto regular.

Definição 4.4 Dado um ponto x^* e o conjunto activo $\mathcal{A}(x^*)$, definido por (4.2), diz que x^* é ponto regular se o conjunto dos gradientes das restrições activas, $\{\nabla h_i(x^*), i \in \mathcal{A}(x^*)\}$, é linearmente independente.

Note-se que se a condição de regularidade se verifica, nenhum dos gradientes das restrições activas pode ser nulo.

Na dedução das condições de optimalidade assume-se sempre que uma solução x^* do problema de optimização (4.1) é um *ponto regular*.

As condições de optimalidade são deduzidas a partir do pressuposto de que se x^* é uma solução local do problema (4.1) então x^* é um minimizante de F ao longo de qualquer curva/arco admissível que passe/parte de x^* .

A condição de regularidade permite enunciar as seguintes condições de optimalidade para um problema de programação não linear (4.1). Designam-se condições necessárias de primeira ordem por estarem relacionadas com as propriedades dos gradientes da função objectivo e das funções de restrição.

Teorema 4.2 (*Condições necessárias de primeira ordem*)

Seja x^* um minimizante local do problema de optimização (4.1). Se x^* é um ponto regular das restrições então existe um vector de multiplicadores de Lagrange λ^* (com componentes λ_i^* , $i = 1, \dots, m + \overline{m}$), tal que as seguintes condições são satisfeitas em (x^*, λ^*)

$$\nabla_x \mathbb{L}(x^*, \lambda^*) = 0, \text{ isto é, } \nabla_x F(x^*) - \nabla_x h(x^*)^T \lambda^* = 0 \quad (4.5)$$

$$h_i(x^*) = 0, \text{ para todo } i = 1, \dots, m, \quad (4.6)$$

$$h_i(x^*) \geq 0, \text{ para todo } i = i, \dots, \overline{m}, \quad (4.7)$$

$$\lambda_i^* \geq 0, \text{ para todo } i = i, \dots, \overline{m}, \quad (4.8)$$

$$\lambda_i^* h_i(x^*) = 0, \text{ para todo } i = 1, \dots, m + \overline{m}. \quad (4.9)$$

Prova. Veja-se no Teorema 12.1 em [8]. ■

As condições de (4.5)-(4.9) são conhecidas por condições de *Karush-Kuhn-Tucker*, ou de modo abreviado condições KKT.

As condições (4.9) designam-se por *condições de complementaridade*. Uma vez que os escalares λ_i^* e $h_i(x^*)$, com $i = 1, \dots, \overline{m}$ são ambos não negativos, implica que $\lambda_i^* h_i(x^*) = 0$ para todo $i = 1, \dots, m + \overline{m}$. Isto significa que, ou a restrição está activa ou o seu correspondente multiplicador de Lagrange é zero. Pelo menos uma das duas é verificada. Estas condições implicam que os multiplicadores de Lagrange que correspondem a restrições não activas são sempre nulos.

A situação em que exactamente uma das duas se verifica (ou $h_i(x^*) = 0$ ou $\lambda_i^* = 0$) mas não ambas, designa-se por *complementaridade estrita*. Neste caso, os multiplicadores de Lagrange que correspondem às restrições activas são sempre positivos (isto é, $\lambda_i^* > 0$, $\forall i \in \{1, \dots, \overline{m}\} \cap \mathcal{A}(x^*)$).

Se o multiplicador de Lagrange que corresponde a uma restrição activa é nulo, a restrição diz-se *degenerada* (ou *fracamente activa*). Esta propriedade é uma característica indesejável uma vez que pode fazer com que os algoritmos progridam muito lentamente, ou falhem.

Note-se que a condição (4.8) e consequentemente o conceito de restrição degenerada só se coloca no âmbito das restrições de desigualdade. Esta condição diz que, numa solução local os multiplicadores de Lagrange associados às restrições de desigualdade não podem ser negativos (um multiplicador negativo indicaria que a função objectivo pode ser melhorada fazendo um pequeno movimento para o interior da restrição correspondente).

Multiplicadores de Lagrange associados a restrições de igualdade podem tomar qualquer valor de \mathbb{R} .

É de salientar que, qualquer solução do problema (4.1) é também uma solução do problema com restrições de igualdade obtido exigindo que as restrições activas naquele ponto sejam exactamente satisfeitas.

Omitindo-se os termos de índices $i \notin \mathcal{A}(x^*)$, a condição (4.5) poderia ser reescrita na forma

$$0 = \nabla_x \mathbb{L}(x^*, \lambda^*) = \nabla F(x^*) - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla h_i(x^*).$$

Para um dado problema (4.1) e um ponto solução x^* , podem existir muitos vectores λ^* para os quais se verificam as condições KKT (4.5)-(4.9). Quando a condição de regularidade se verifica o óptimo λ^* é único.

Os resultados seguintes podem ser consultados em [7].

Lema 4.1 *As seguintes afirmações são verdadeiras.*

- i. *Se x^* é um minimizante local de F então os vectores $p \in \mathbb{R}^n$ tangentes às curvas/arcs admissíveis que passam/partem em x^* verificam a condição*

$$p^T \nabla F(x^*) \geq 0.$$

- ii. *Se $p \in \mathbb{R}^n$ é um vector tangente à curva/arco admissível que passa/parte de x^* , então*

$$p^T \nabla h_i(x^*) = 0, \text{ para } i = 1, \dots, m; \quad p^T \nabla h_i(x^*) \geq 0, \text{ para todo } i \in \{1, \dots, \overline{m}\} \cap \mathcal{A}(x^*).$$

Prova. Veja-se na Secção 14.7 em [7]. ■

Considere-se ainda a seguinte definição.

Definição 4.5 *Dado um ponto x^* , o conjunto $\mathcal{C}_1(x^*)$ é definido por*

$$\mathcal{C}_1(x^*) = \left\{ p \in \mathbb{R}^n : \begin{array}{ll} p^T \nabla h_i(x^*) = 0, & \text{para todo } i = 1, \dots, m; \\ p^T \nabla h_i(x^*) \geq 0, & \text{para todo } i \in \{1, \dots, \overline{m}\} \cap \mathcal{A}(x^*). \end{array} \right\}.$$

$\mathcal{C}_1(x^*)$ é o conjunto de todos os vectores tangentes às curvas/arcs admissíveis que passam/partem de x^* . Assume-se por conveniência que $0 \in \mathcal{C}_1(x^*)$. Assim, se $p \in \mathcal{C}_1(x^*)$ então $ap \in \mathcal{C}_1(x^*)$ para qualquer escalar não negativo a . Um conjunto com esta propriedade diz-se um *cone*, sendo $\mathcal{C}_1(x^*)$ por vezes conhecido como o *cone tangente* em x^* .

4.3.2 Condições de optimalidade de 2ª ordem

As condições necessárias de primeira ordem mostram que as primeiras derivadas parciais de F e h_i estão relacionadas no ponto x^* . Quando estas condições se verificam, um movimento ao longo de qualquer vector $p \in \mathcal{C}_1(x^*)$ ou incrementa a primeira derivada da aproximação à função objectivo (que é, $p^T \nabla F(x^*) > 0$) ou mantém o mesmo valor (que é, $p^T \nabla F(x^*) = 0$). Para uma direcção $p \in \mathcal{C}_1(x^*)$ que verifica $p^T \nabla F(x^*) = 0$, não é possível determinar, a partir apenas da informação das primeiras derivadas, se um movimento ao longo desta direcção (movimento admissível) vai incrementar ou diminuir o valor da função objectivo. É pois necessário analisar as segundas derivadas. Na sua essência, as condições de segunda ordem dizem respeito à curvatura da função Lagrangeana ao longo destas direcções.

Dado $\mathcal{C}_1(x^*)$, a partir da definição 4.5 e para algum vector de multiplicadores de Lagrange λ^* que satisfaz as condições KKT (4.5)-(4.9), define-se um subconjunto de $\mathcal{C}_1(x^*)$ por

$$\mathcal{C}_2(x^*) = \left\{ p \in \mathcal{C}_1(x^*) : \begin{array}{ll} p^T \nabla h_i(x^*) = 0, & \text{para todo } i = 1, \dots, m; \\ p^T \nabla h_i(x^*) = 0, & \text{para todo } i \in \{1, \dots, \overline{m}\} \cap \mathcal{A}(x^*) \text{ com } \lambda_i^* > 0 \end{array} \right\}. \quad (4.10)$$

Note-se que foram desprezadas as restrições de desigualdade que são não activas nos novos pontos admissíveis, uma vez que não influenciam as condições de optimalidade, e as que são degeneradas.

O teorema seguinte estabelece uma condição necessária que envolve as segundas derivadas: se x^* é uma solução local, então a curvatura da Lagrangeana ao longo das direcções em $\mathcal{C}_2(x^*)$ tem que ser não negativa.

Teorema 4.3 (*Condições necessárias de 2ª ordem*)

Seja x^ um minimizante local do problema de optimização (4.1) que é ponto regular. Seja λ^* um vector de multiplicadores de Lagrange que verifica as condições KKT (4.5)-(4.9). Então*

$$p^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) p \geq 0, \text{ para todo } p \in \mathcal{C}_2(x^*). \quad (4.11)$$

Prova. Veja-se no Teorema 12.5 em [8]. ■

As condições suficientes são condições em F e h_i , para $i = 1, \dots, m + \overline{m}$, que garantem que x^* é uma solução local do problema (4.1). As condições suficientes de segunda ordem enunciadas no teorema seguinte, apesar de semelhantes às condições necessárias de segunda ordem, não exigem que x^* seja ponto regular e a desigualdade (4.11) é substituída por uma desigualdade estrita

Teorema 4.4 (*Condições suficientes de 2ª ordem*)

Suponha que para algum ponto admissível $x^* \in \mathbb{R}^n$, existe um vector de multiplicadores de Lagrange λ^* tal que as condições KKT (4.5)-(4.9) se verificam. Suponha também que

$$p^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) p > 0, \text{ para todo } p \in \mathcal{C}_2(x^*), p \neq 0. \quad (4.12)$$

Então x^* é uma solução local estrita para (4.1).

Prova. Veja-se no Teorema 12.6 em [8]. ■

Se o multiplicador de Lagrange é zero, é impossível prever, a partir da informação de 1ª ordem, se um pequeno movimento para o interior da restrição causa um incremento no valor da função objectivo. Assim, no caso de uma restrição degenerada, são exigidas condições mais rígidas sobre a Hessiana para garantir que um ponto é minimizante local [8].

Se o minimizante x^* é ponto regular e a complementaridade estrita se verifica (isto é, não existem restrições degeneradas) então a definição de (4.10) de $\mathcal{C}_2(x^*)$ reduz-se a

$$\mathcal{C}_2(x^*) = \text{Nuc}[\nabla h_i(x^*)]_{i \in \mathcal{A}(x^*)} = \text{Nuc}[\overline{\nabla h}(x^*)]$$

em que $\overline{\nabla h}$ representa a matriz do Jacobiano das restrições activas em x^* . Ou seja, $\mathcal{C}_2(x^*)$ é o núcleo da matriz cujas linhas são os gradientes das restrições activas em x^* . Seja $Z(x^*)_{n \times (n-\overline{m})}$ uma matriz cujas colunas formam uma base para núcleo¹ do Jacobiano das restrições activas em x^* (\overline{m} é o número de restrições activas em x^*). Qualquer vector $p \in \mathcal{C}_2(x^*)$ pode ser escrito na forma $p = Zv$. Assim, a condição (4.11) do Teorema 4.3 pode ser escrita na forma

$$v^T Z^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) Z v \geq 0, \quad \text{para todo } v,$$

ou, mais sucintamente,

$$Z^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) Z \quad \text{é semidefinida positiva.}$$

Similarmente, a condição (4.12) do Teorema 4.4 pode ser escrita na forma

¹**Definição:** Seja A uma matriz de ordem $m \times n$. Denota-se o núcleo de A por

$$\mathcal{N}(A) = \{p \in \mathbb{R}^n : Ap = 0\}.$$

$Z^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) Z$ é definida positiva.

A matriz $Z^T \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) Z$ designa-se por matriz Hessiana reduzida, da Hessiana da Lagrangeana.

4.4 Exercícios

1. Verifique se o ponto $x^* = (2.5, -1.5, -1)^T$ satisfaz as condições de optimalidade do problema

$$\begin{aligned} \text{minimizar} \quad & x_1^2 - 2x_1 + x_2^2 - x_3^2 + 4x_3 \\ \text{s.a} \quad & x_1 - x_2 + 2x_3 = 2. \end{aligned}$$

2. Considere o problema

$$\begin{aligned} \text{minimizar} \quad & x_1^2 + x_1^2 x_3^2 + 2x_1 x_2 + x_2^4 + 8x_2 \\ \text{s.a} \quad & 2x_1 + 5x_2 + x_3 = 3. \end{aligned}$$

Verifique se os pontos $(0, 0, 2)^T$, $(0, 0, 3)^T$ e $(1, 0, 1)^T$

- (a) são pontos estacionários da Lagrangeana.
 (b) são minimizantes locais, maximizantes locais ou pontos sela.

3. Verifique se o ponto $x^* = (1, 1)^T$ satisfaz as condições de optimalidade do problema

$$\begin{aligned} \text{minimizar} \quad & x_1^2 + x_2^2 \\ \text{s.a} \quad & x_1 + x_2 = 2. \end{aligned}$$

4. Verifique se o ponto $x^* = (1, 0, 0)^T$ satisfaz as condições de optimalidade do problema

$$\begin{aligned} \text{minimizar} \quad & x_1^4 x_2^2 + x_1^2 x_3^4 + \frac{1}{2} x_1^2 + x_1 x_2 + x_3 \\ \text{s.a} \quad & x_1 + x_2 + x_3 = 1. \end{aligned}$$

5. Considere os seguintes problemas

(a) minimizar $x_1^2 - x_2^2$
 s.a $x_1^2 + 2x_2^2 = 4$.

(b) minimizar $(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$
 s.a. $x_1 - 2 = 0$
 $x_3^2 + x_4^2 - 2 = 0$

(c) minimizar $2(x_1^2 + x_2^2 - 1) - x_1$
 s.a $x_1^2 + x_2^2 - 1 = 0$.

Calcule os pontos estacionários da Lagrangeana associada a cada um dos problemas e indique quais são os minimizantes locais.

6. Resolva os seguintes problemas

$$\begin{aligned} \text{(a) minimizar } & x_1^3 - x_2^3 - 2x_1^2 - x_1 + x_2 \\ \text{s.a. } & -x_1 - 2x_2 \geq -2 \\ & x_1 \geq 0 \\ & x_2 \geq 0. \end{aligned}$$

$$\begin{aligned} \text{(b) minimizar } & \frac{1}{2}x_1^2 + x_2^2 \\ \text{s.a. } & 2x_1 + x_2 \geq 2 \\ & x_1 - x_2 \leq 1 \\ & x_1 \geq 0. \end{aligned}$$

$$\begin{aligned} \text{(c) minimizar } & -x_1^2 + x_2^2 - x_1x_2 \\ \text{s.a. } & 2x_1 - x_2 \geq 2 \\ & x_1 + x_2 \leq 4 \\ & x_1 \geq 0. \end{aligned}$$

Capítulo 5

Programação quadrática sequencial

Um dos métodos mais eficazes para optimização não linear com restrições gera os passos através da resolução de subproblemas quadráticos. Este método, conhecido por programação quadrática sequencial (PQS), pode ser combinado com uma técnica de procura unidimensional ou com regiões de confiança, para efeitos de globalização do algoritmo, e é apropriado tanto para problemas de pequena como de grande dimensão.

Pretende-se aplicar um método PQS com procura unidimensional para a resolução de problemas de optimização não linear com restrições de igualdade. A apresentação do método (PQS) será feita em duas etapas. Primeiro, apresenta-se um algoritmo local que motiva a aproximação PQS e que permite introduzir o cálculo da direcção de procura e as técnicas quasi-Newton de aproximação à Hessiana. A seguir, considera-se um método PQS com procura unidimensional, prático, que converge a partir de pontos iniciais distantes.

5.1 Método da programação quadrática sequencial

Considere-se a seguinte formulação do problema de optimização não linear com restrições de igualdade

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimizar}} && F(x) \\ & \text{sujeito a} && h(x) = 0. \end{aligned} \tag{5.1}$$

A função Lagrangeana associada ao problema (5.1) é

$$\mathbb{L}(x, \lambda) = F(x) - \lambda^T h(x) \tag{5.2}$$

em que $\lambda \in \mathbb{R}^m$ é o vector dos multiplicadores de Lagrange associado às restrições em (5.1). As condições de Karush-Kuhn-Tucker de primeira ordem para um minimizante são

$$\begin{cases} \nabla_x \mathbb{L} = 0 \\ \nabla_\lambda \mathbb{L} = 0 \end{cases}$$

ou seja,

$$\begin{cases} \nabla F(x) - \nabla h(x)^T \lambda = 0 \\ \nabla h(x) = 0 \end{cases} \quad (5.3)$$

O sistema (5.3) é não linear de $n + m$ equações nas $n + m$ incógnitas x e λ . Se $\nabla h(x^*)$ tiver característica completa por linhas, qualquer par (x^*, λ^*) , em que x^* é a solução do problema com restrições de igualdade (5.1), satisfaz (5.3). Considerando o método de Newton na resolução de (5.3), e para simplificar a notação, considere-se o vector das funções F_1 e F_2 definido por

$$\begin{cases} F_1(x, \lambda) = \nabla F(x) - \nabla h(x)^T \lambda = 0 \\ F_2(x, \lambda) = \nabla h(x) = 0. \end{cases}$$

Para implementar o método de Newton, precisa-se de formar a matriz do Jacobiano de $\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$, que é dada por

$$\begin{pmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial \lambda} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial \lambda} \end{pmatrix} \in R^{(n+m) \times (n+m)}$$

ou seja

$$\begin{pmatrix} W(x, \lambda) & -\nabla h^T(x) \\ \nabla h(x) & 0 \end{pmatrix}$$

onde $W(x, \lambda)$ denota a Hessiana da Lagrangeana dada por

$$W(x, \lambda) \equiv \nabla_{xx}^2 \mathbb{L}(x, \lambda) = \nabla^2 F(x) - \sum_{i=1}^m \lambda_i \nabla^2 h_i(x). \quad (5.4)$$

As equações para as direcções de procura são

$$\begin{pmatrix} W(x, \lambda) & -\nabla h^T(x) \\ \nabla h(x) & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = - \begin{bmatrix} \nabla F(x) - \nabla h(x)^T \lambda \\ h(x) \end{bmatrix} \quad (5.5)$$

em que Δx , $\Delta \lambda$ definem o vector das direcções de Newton.

A partir de um ponto inicial (x_1, λ_1) , o algoritmo prossegue iterativamente ao longo de uma sucessão de pontos determinada pelas direcções de procura acima descritas:

$$\begin{aligned} x_{k+1} &= x_k + \Delta x_k \\ \lambda_{k+1} &= \lambda_k + \Delta \lambda_k, \quad k = 1, 2, \dots \end{aligned} \quad (5.6)$$

Este processo iterativo, é conhecido por método de Newton-Lagrange, e está bem bem definido quando a matriz dos coeficientes do sistema (5.5) é não singular. A não singularidade é uma consequência das seguintes hipóteses:

- (a) o Jacobiano das restrições $\nabla h(x_k)$ tem característica completa por linhas;
- (b) a matriz $W(x_k, \lambda_k)$ é definida positiva no espaço tangente das restrições, isto é, $p^T W(x_k, \lambda_k) p > 0$ para todo $p \neq 0$ tal que $\nabla h(x_k) p = 0$.

A hipótese (a) diz respeito à condição de regularidade apresentada no Capítulo 4 (veja-se a Definição 4.4), a qual se assume ao longo deste capítulo. A segunda condição verifica-se sempre que (x, λ) esteja perto do óptimo (x^*, λ^*) e a condição suficiente de segunda ordem seja satisfeita na solução (veja-se Teorema 4.4).

Se estas hipóteses são verificadas, é possível mostrar que o processo iterativo definido por (5.5) e (5.6) converge quadraticamente e é um bom algoritmo para a resolução de problemas com restrições de igualdade, desde que o ponto inicial esteja perto de x^* .

De seguida, vai mostrar-se a analogia bem como a equivalência entre este método e o método da programação quadrática sequencial.

A ideia central da PQS é aproximar (5.1), no iterando actual x_k , por um subproblema de programação quadrática e usar o minimizante deste subproblema para definir um novo iterando x_{k+1} . O objectivo é, assim, o de conceber o subproblema quadrático de modo a que produza um bom passo para o problema de optimização com restrições, que lhe está subjacente, e que o algoritmo PQS tenha boas propriedades de convergência e bom desempenho prático.

Considere-se, então, com base na aproximação (x_k, λ_k) o seguinte programa quadrático

$$\begin{aligned} & \underset{\Delta \in \mathbb{R}^n}{\text{minimizar}} \quad \frac{1}{2} \Delta^T W_k \Delta + \nabla F_k^T \Delta \\ & \text{sujeito a} \quad \nabla h_k \Delta + h_k = 0 \end{aligned} \tag{5.7}$$

onde W_k , ∇F_k , ∇h_k e h_k denotam respectivamente $W(x_k, \lambda_k)$, $\nabla F(x_k)$, $\nabla h(x_k)$ e $h(x_k)$.

A função Lagrangeana associada ao problema (5.7) é

$$\mathbb{L}(\Delta, \pi) = \frac{1}{2} \Delta^T W_k \Delta + \nabla F_k^T \Delta - \pi^T (\nabla h_k \Delta + h_k)$$

em que $\pi \in \mathbb{R}^m$ é o vector dos multiplicadores de Lagrange associado às restrições em (5.7). As condições KKT de primeira ordem para um minimizante, na iteração k , são

$$\begin{cases} W_k \Delta_k + \nabla F_k - \nabla h_k^T \pi_k = 0 \\ \nabla h_k \Delta_k + h_k = 0 \end{cases}$$

ou seja,

$$\begin{pmatrix} W_k & -\nabla h_k^T \\ \nabla h_k & 0 \end{pmatrix} \begin{pmatrix} \Delta_k \\ \pi_k \end{pmatrix} = - \begin{pmatrix} \nabla F_k \\ h_k \end{pmatrix}. \quad (5.8)$$

Se as hipóteses (a) e (b) se verificam, o problema (5.7) tem uma única solução (Δ_k, π_k) que satisfaz (5.8).

Uma observação importante é que Δ_k e π_k podem ser identificados com a solução das equações de Newton (5.5). Considerando a iteração k e subtraindo $\nabla h_k^T \lambda_k$ a ambos os lados da primeira equação de (5.5), tem-se

$$W_k \Delta x_k - \nabla h_k^T (\Delta \lambda_k + \lambda_k) = -\nabla F_k$$

e por (5.6) vem

$$W_k \Delta x_k - \nabla h_k^T \lambda_{k+1} = -\nabla F_k.$$

Portanto, o sistema (5.5) é equivalente a

$$\begin{pmatrix} W_k & -\nabla h_k^T \\ \nabla h_k & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \lambda_{k+1} \end{pmatrix} = - \begin{pmatrix} \nabla F_k \\ h_k \end{pmatrix}. \quad (5.9)$$

Portanto, pela não singularidade da matriz dos coeficientes, tem-se que

$$\Delta_k = \Delta x_k$$

e

$$\lambda_{k+1} = \pi_k.$$

Esta relação é referida como a *equivalência entre PQS e o método de Newton-Lagrange*. Se as hipóteses (a) e (b) se verificam em x_k , então o novo iterando (x_{k+1}, λ_{k+1}) pode ser definido como a solução do programa quadrático (5.7) ou como o iterando gerado pelo método de Newton (5.5)-(5.6) aplicado às condições de optimalidade do problema. Estas interpretações alternativas são importantes. Do ponto de vista do método de Newton facilita a análise de convergência, enquanto que a PQS permite deduzir algoritmos práticos e estender a técnica para os problemas com restrições de desigualdade (veja-se em [8]).

É fácil estabelecer um resultado de convergência local para o algoritmo de PQS, uma vez que se sabe que é equivalente a aplicar o método de Newton às condições de optimalidade de primeira ordem de (5.1). Mais concretamente, se as hipóteses (a) e (b) se verificam numa solução (x^*, λ^*) de (5.1), se F e h são duas vezes diferenciáveis com as segundas derivadas contínuas à Lipschitz, e se o ponto inicial (x_1, λ_1) está suficientemente perto de (x^*, λ^*) , então os iterandos gerados pelo algoritmo de PQS convergem quadraticamente para (x^*, λ^*) ([8], Secção 18.10).

Portanto, com o programa quadrático definido por (5.7) é possível provar convergência quadrática para o método da programação quadrática sequencial. Note-se que na função objectivo do programa quadrático (5.7), pode substituir-se o termo linear $\nabla F_k^T \Delta$ por $\nabla_x \mathbb{L}(x_k, \lambda_k)^T \Delta$, uma vez que a restrição de (5.7) torna as duas escolhas equivalentes. Nesta situação, a função objectivo do programa quadrático (5.7) é uma aproximação quadrática da função Lagrangeana. Esta aproximação é feita numa vizinhança pequena, ao redor do ponto x_k . Vejamos, considerando a expansão em série de Taylor até à segunda ordem da função \mathbb{L} , ao redor do ponto x_k , obtém-se por (5.2), (5.3) e (5.4) e usando a restrição de (5.7), a seguinte forma quadrática em $\Delta \in R^n$

$$\begin{aligned}
 \mathbb{L}(x_k + \Delta, \lambda_k) &\approx \mathbb{L}(x_k, \lambda_k) + \nabla_x \mathbb{L}(x_k, \lambda_k)^T \Delta + \frac{1}{2} \Delta^T \nabla_{xx}^2 \mathbb{L}(x_k, \lambda_k)^T \Delta \\
 &= F(x_k) - \lambda_k^T h(x_k) + \nabla F(x_k)^T \Delta - \lambda_k^T \nabla h(x_k) \Delta + \frac{1}{2} \Delta^T W(x_k, \lambda_k)^T \Delta \\
 &= F(x_k) - \lambda_k^T h(x_k) + \nabla F(x_k)^T \Delta - \lambda_k^T h(x_k) + \frac{1}{2} \Delta^T W(x_k, \lambda_k)^T \Delta \\
 &= F(x_k) + \nabla F(x_k)^T \Delta + \frac{1}{2} \Delta^T W(x_k, \lambda_k)^T \Delta \\
 &\approx \nabla F(x_k)^T \Delta + \frac{1}{2} \Delta^T W(x_k, \lambda_k)^T \Delta
 \end{aligned}$$

em que se ignorou o escalar $F(x_k)$ da forma quadrática dado que não afecta a solução do problema, isto é, o valor de Δ que minimiza $\mathbb{L}(x_k + \Delta, \lambda_k)$.

Da mesma forma, definiu-se uma aproximação linear em $\Delta \in R^n$ das funções de restrição, ao redor do ponto x_k ,

$$h(x_k + \Delta) \approx h(x_k) + \nabla h(x_k) \Delta.$$

Portanto nesta situação, o problema quadrático definido por (5.7) conduz a uma motivação alternativa do método PQS. O programa não linear (5.1) é substituído pelo problema que consiste em minimizar a Lagrangeana sujeito às restrições de igualdade de (5.1). Fazendo uma aproximação quadrática da Lagrangeana e uma aproximação linear das restrições obtém-se (5.7).

5.2 Métodos PQS práticos

Em termos práticos, um método PQS tem de ser capaz de convergir a partir de qualquer ponto inicial e para qualquer problema não convexo. Assim, vão ser apresentadas nesta secção as modificações que devem ser introduzidas na estratégia PQS local para alcançar estes objectivos.

Em analogia com o que se passa em optimização sem restrições, uma iteração do método de Newton para minimizar uma função F toma um passo em direcção ao mínimo do modelo quadrático

$$m_k(\Delta) = F_k + \nabla F_k^T \Delta + \frac{1}{2} \Delta^T \nabla^2 F_k \Delta.$$

Esta abordagem é útil na vizinhança da solução onde a Hessiana, $\nabla^2 F(x_k)$, é normalmente definida positiva e o modelo quadrático tem um mínimo bem definido. Contudo, quando x_k está longe da solução, a função m_k pode não ser convexa. Os métodos de procura unidimensional modificam a Hessiana de $m_k(\Delta)$ de modo a torná-la definida positiva (possivelmente substituindo-a por uma aproximação quasi-Newton, B_k), garantindo, assim, que Δ_k é uma direcção descendente para a função objectivo F .

Nos métodos PQS usam-se estratégias idênticas para a sua globalização. Se W_k é definida positiva no espaço tangente das restrições (hipótese (b)), o subproblema quadrático (5.7) tem uma única solução. No entanto, quando W_k não é definida positiva, os métodos PQS de procura unidimensional substituem W_k por uma aproximação definida positiva, B_k , ou modificam W_k directamente durante o processo de factorização. Uma terceira possibilidade consiste em definir W_k como a Hessiana de uma função Lagrangeana aumentada que possui certas propriedades de convexidade. Em qualquer destes casos, o subproblema (5.7) ficará bem definido.

A técnica usada para resolver os subproblemas de procura unidimensional tem um grande impacto na eficiência e robustez dos métodos PQS, em particular nos problemas de grandes dimensões.

Uma outra questão importante no desenvolvimento dos métodos PQS é a escolha de uma função mérito que guie os algoritmos para a solução do problema. Em optimização sem restrições a função mérito é a própria função objectivo, F , que é fixa durante o processo de minimização. Os métodos PQS podem usar qualquer das funções mérito apresentadas em [8, Capítulo 15], embora os seus parâmetros possam precisar de ser ajustados em algumas iterações para garantir que a direcção obtida, a partir do subproblema, é uma direcção descendente para esta função. As regras de actualização destes parâmetros requerem algum cuidado, uma vez que têm influência sobre o comportamento prático dos métodos PQS.

Neste capítulo, vão ser apresentadas algumas estratégias para definir um algoritmo PQS prático para a resolução de problemas de optimização não linear com restrições de igualdade. Começa-se por discutir a técnica usada para resolver o subproblema quadrático (5.7) e analisar o seu efeito na forma da iteração PQS. De seguida, consideram-se várias formulações do modelo quadrático que garantem a sua adequação no contexto da procura unidimensional. Consideram-se o caso em que W_k é a Hessiana exacta da Lagrangeana e o caso em que é uma aproximação quasi-Newton. Apresentam-se duas funções mérito e mostra-se que as direcções PQS são direcções descendentes para estas funções. Deste modo, ficam estabelecidas as condições para se apresentar um método PQS prático de

procura unidimensional.

5.3 Cálculo da direcção de procura

Considere-se o programa quadrático com restrições de igualdade (5.7). Assumiu-se atrás que a Hessiana deste modelo é definida como sendo a Hessiana da Lagrangeana $W_k = \nabla_{xx}^2 \mathbb{L}(x_k, \lambda_k)$. Na Secção 5.4 esta matriz será substituída por uma aproximação quasi-Newton, no entanto, a discussão desta secção é independente desta escolha.

Já foi referido que, se as hipóteses (a) e (b) se verificam, a solução de (5.7) é dada pelo sistema KKT (5.9). Existem várias técnicas para resolver este sistema. Neste capítulo apresentam-se dois métodos para a resolução do sistema KKT. A resolução directa do sistema KKT pelo método de Gauss com escolha parcial de pivot e a resolução pelo método dual ou espaço-imagem. Em alternativa, pode usar-se uma das técnicas referidas em [8].

Método dual ou espaço-imagem

Se W_k é definida definida positiva, pode-se separar o sistema (5.9) nas suas equações

$$\begin{aligned} W_k \Delta x_k - \nabla h_k^T \lambda_{k+1} &= -\nabla F_k \\ \nabla h_k \Delta x_k &= -h_k \end{aligned}$$

e multiplicar ambos os membros da primeira equação por $\nabla h_k W_k^{-1}$, obtendo-se

$$\nabla h_k \Delta x_k - \nabla h_k W_k^{-1} \nabla h_k^T \lambda_{k+1} = -\nabla h_k W_k^{-1} \nabla F_k.$$

Substituindo $\nabla h_k \Delta x_k$ por $-h_k$ nesta equação obtém-se

$$\nabla h_k W_k^{-1} \nabla h_k^T \lambda_{k+1} = \nabla h_k W_k^{-1} \nabla F_k - h_k.$$

Assim, a resolução do sistema KKT (5.9) é dada pela seguinte ordem de resolução dos dois sistemas seguintes, para obter-se λ_{k+1} e Δx_k :

$$\nabla h_k W_k^{-1} \nabla h_k^T \lambda_{k+1} = \nabla h_k W_k^{-1} \nabla F_k - h_k.$$

$$W_k \Delta x_k = -\nabla F_k + \nabla h_k^T \lambda_{k+1}.$$

Este método é particularmente eficaz para métodos quasi-Newton que explicitamente aproximam W_k^{-1} por uma matriz definida positiva H_k .

5.4 A Hessiana do modelo quadrático

A equivalência entre o método PQS e o método de Newton aplicado às condições de optimalidade de primeira ordem (5.3) é baseada no facto de que W_k é a Hessiana da Lagrangeana. Esta escolha conduz a uma taxa de convergência quadrática sob hipóteses razoáveis e muitas vezes também produz progresso rápido quando os iterandos estão longe da solução. No entanto, esta matriz é formada pelas segundas derivadas da função objectivo e das restrições, que nem sempre são fáceis de calcular. Além disso, W_k pode não ser definida positiva no núcleo das restrições (hipótese (b)). Relembrar que, para o problema (5.1)

$$W_k = \nabla_{xx}^2 \mathbb{L}(x_k, \lambda_k) = \nabla^2 F(x_k) - \sum_{i=1}^m \lambda_{i,k} \nabla^2 h_i(x_k). \quad (5.10)$$

5.4.1 Aproximações quasi-Newton

A primeira ideia que surge é aproximar a Hessiana da Lagrangeana $\nabla_{xx}^2 \mathbb{L}(x_k, \lambda_k)$ por uma matriz de actualização quasi-Newton B_k .

Uma vez que a fórmula de actualização que melhores resultados tem dado no contexto da optimização sem restrições, para aproximar Hessianas, é a de BFGS, vai aplicar-se esta mesma estratégia a este caso. Esta fórmula de actualização conserva as matrizes simétricas e, sob certas condições, garante que as aproximações sejam definidas positivas.

Assim, na iteração k ,

$$W_k = \nabla_{xx}^2 \mathbb{L}(x_k, \lambda_k) \approx B_k.$$

A fórmula de actualização para B_k , que resulta do passo do iterando k para o iterando $k+1$, usa os vectores s_k e y_k , os quais são definidos como se segue:

$$s_k = x_{k+1} - x_k \quad y_k = \nabla_x \mathbb{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathbb{L}(x_k, \lambda_{k+1}). \quad (5.11)$$

A seguir, calcula-se a nova aproximação B_{k+1} usando a fórmula de actualização BFGS (2.22).

Note-se que, esta estratégia pode ser vista como uma aplicação da actualização quasi-Newton ao caso em que a função objectivo é dada pela função Lagrangeana $\mathbb{L}(x, \lambda)$ (com λ fixo). Estas definições de imediato revelam a força e a fragilidade destas aproximações.

Se $\nabla_{xx}^2 \mathbb{L}$ é definida positiva na região onde se encontra o minimizante, as aproximações quasi-Newton, B_k , reflectem alguma informação da curvatura do problema, e o processo iterativo é robusto e converge rapidamente, como quando aplicadas no contexto dos problemas de optimização sem restrições. Se, no entanto, $\nabla_{xx}^2 \mathbb{L}$ contém valores próprios negativos, então as aproximações quasi-Newton BFGS, ao aproximá-la por uma matriz definida positiva, podem não ser eficazes. De facto, estas actualizações exigem que

s_k e y_k satisfaçam a condição de curvatura $s_k^T y_k > 0$, o que pode não acontecer, quando s_k e y_k são dados por (5.11), mesmo na vizinhança da solução.

Apresenta-se a seguir duas estratégias para ultrapassar esta dificuldade.

Estratégia de skipping

Esta estratégia consiste em não actualizar B_k se a condição

$$s_k^T y_k \geq \gamma s_k^T B_k^T s_k$$

não se verificar, onde γ é um parâmetro positivo ($\gamma = 10^{-2}$, por exemplo). Esta estratégia, conhecida na literatura como *skipping* [8], já foi implementada em alguns algoritmos PQS, juntamente com a fórmula BFGS em optimização não linear, e mostrou ter um bom desempenho em muitos problemas. Noutros problemas, no entanto, revelou fraco desempenho, não podendo portanto ser vista como uma regra de propósito geral.

Estratégia damped

A estratégia damped consiste em modificar a definição de y_k de modo a garantir que a actualização BFGS esteja sempre bem definida. A aplicação desta à fórmula de actualização BFGS (2.22), origina o seguinte algoritmo.

Algoritmo 5.1 (*Actualização BFGS damped para PQS*)

Definir s_k e y_k como em (5.11) e fazer

$$r_k = \theta_k y_k + (1 - \theta_k) B_k s_k,$$

onde o escalar θ_k é definido por

$$\theta_k = \begin{cases} 1 & \text{se } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k} & \text{se } s_k^T y_k < 0.2 s_k^T B_k s_k. \end{cases}$$

Actualizar B_k como se segue:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{r_k r_k^T}{s_k^T r_k}. \quad (5.12)$$

A fórmula (5.12) é simplesmente a fórmula BFGS, com y_k substituído por r_k . Esta fórmula garante que a condição de curvatura se verifica ao longo do processo iterativo. Apresenta-se, de seguida, esta demonstração. Se $\theta_k \neq 1$ então

$$\theta_k = \frac{0.8 b_k}{b_k - a_k}$$

onde $a_k = s_k^T y_k$ e $b_k = s_k^T B_k s_k$. Assim sendo, tem-se

$$r_k = \frac{0.8b_k}{b_k - a_k} y_k + \left(1 - \frac{0.8b_k}{b_k - a_k}\right) B_k s_k$$

e portanto

$$\begin{aligned} s_k^T r_k &= \frac{0.8b_k}{b_k - a_k} s_k^T y_k + \left(1 - \frac{0.8b_k}{b_k - a_k}\right) s_k^T B_k s_k \\ &= \frac{0.8b_k a_k}{b_k - a_k} + \frac{b_k^2 - a_k b_k - 0.8b_k^2}{b_k - a_k} \\ &= \frac{0.2b_k^2 - 0.2a_k b_k}{b_k - a_k} \\ &= 0.2b_k \equiv 0.2s_k^T B_k s_k > 0 \end{aligned}$$

como se queria demonstrar.

Note-se que, quando $\theta_k = 0$, tem-se em ambos os casos $B_{k+1} = B_k$, e quando $\theta_k = 1$, (5.12) é exactamente a fórmula de actualização BFGS não modificada. Um valor de $\theta_k \in (0, 1)$ produz uma matriz entre a aproximação B_k e a matriz obtida pela fórmula de actualização BFGS não modificada.

Um problema que continua a existir, mesmo tendo-se incorporado estas estratégias na fórmula de actualização BFGS, é que esta actualização falha nos problemas em que a Hessiana da Lagrangeana não é definida positiva. Uma solução para este problema consiste em modificar directamente a Hessiana da Lagrangeana através da adição de termos à função Lagrangeana, com a finalidade de garantir a positividade da Hessiana.

5.4.2 A Hessiana da Lagrangeana aumentada

Considere-se a função Lagrangeana aumentada definida por

$$\mathbb{L}_A(x, \lambda; \bar{\beta}) = F(x) - \lambda^T h(x) + \frac{\bar{\beta}}{2} \|h(x)\|_2^2, \quad (5.13)$$

para algum escalar positivo $\bar{\beta}$. Num minimizante (x^*, λ^*) , que satisfaz as condições suficientes de segunda ordem, a Hessiana desta função, dada por

$$\nabla_{xx}^2 \mathbb{L}_A(x^*, \lambda^*; \bar{\beta}) = \nabla_{xx}^2 \mathbb{L}(x^*, \lambda^*) + \bar{\beta} \nabla h^T(x^*) \nabla h(x^*) \quad (5.14)$$

é definida positiva para todo o $\bar{\beta}$ que seja maior do que um certo valor $\bar{\beta}^*$ [8]. Note-se que, o último termo em (5.14) adiciona curvatura positiva à Lagrangeana no espaço gerado pelas colunas de $\nabla h^T(x)$, deixando inalterada a curvatura sobre o espaço núcleo

de $\nabla h(x)$. Pode escolher-se agora a matriz W_k do subproblema quadrático (5.1) como sendo $\nabla_{xx}^2 \mathbb{L}_A(x_k, \lambda_k; \bar{\beta})$, ou alguma aproximação quasi-Newton B_k^A para esta matriz. Para valores suficientemente grandes de $\bar{\beta}$, esta Hessiana é definida positiva e pode usar-se directamente nos métodos PQS de procura unidimensional.

No entanto, existem algumas dificuldades com esta estratégia. O valor de referência, $\bar{\beta}^*$, depende de quantidades que não são normalmente conhecidas, tais como limites para as segundas derivadas das funções do problema, sendo portanto difícil de escolher um valor apropriado para $\bar{\beta}$. Valores para $\bar{\beta}$ muito grandes podem fazer com que o último termo de (5.14) domine a Hessiana da Lagrangeana original, levando a um fraco desempenho prático. Se $\bar{\beta}$ for muito pequeno, a Hessiana da Lagrangeana aumentada pode não ser definida positiva, \mathbb{L}_A pode não ser convexa, e a condição de curvatura $s_k^T y_k > 0$ não é verificada.

Uma outra alternativa consiste em usar a fórmula de actualização BFGS para aproximar $\nabla_{xx}^2 \mathbb{L}_A$. Assim, na iteração k ,

$$\nabla_{xx}^2 \mathbb{L}_A(x_k, \lambda_k; \bar{\beta}) \approx B_k^A.$$

Para este caso, a fórmula actualização da matriz B_k^A usam os vectores s_k^A e y_k^A , que são definidos como se segue:

$$s_k^A = s_k = x_{k+1} - x_k \quad (5.15)$$

$$\begin{aligned} y_k^A &= \nabla_x \mathbb{L}_A(x_{k+1}, \lambda_{k+1}; \bar{\beta}) - \nabla_x \mathbb{L}_A(x_k, \lambda_{k+1}; \bar{\beta}) \\ &= \nabla_x \mathbb{L}(x_{k+1}, \lambda_{k+1}) + \bar{\beta} \nabla h^T(x_{k+1}) h(x_{k+1}) - \nabla_x \mathbb{L}(x_k, \lambda_{k+1}) - \\ &\quad - \bar{\beta} \nabla h^T(x_k) h(x_k) \\ &= y_k + \bar{\beta} \nabla h^T(x_{k+1}) h(x_{k+1}) - \bar{\beta} \nabla h^T(x_k) h(x_k). \end{aligned}$$

Pode mostrar-se que na vizinhança da solução existe um valor mínimo de $\bar{\beta}$ (valor de referência) que garante positividade de $s_k^T y_k^A$. Na fórmula de actualização BFGS o vector y_k é substituído por y_k^A . Como existem poucas experiências relacionadas com o tipo de aproximação aqui descrito, não é possível concluir sobre a robustez e eficiência dos correspondentes métodos PQS.

5.5 Funções mérito e descendência

Para garantir que o método PQS com procura unidimensional convirja a partir de qualquer ponto inicial, deve introduzir-se no algoritmo uma função mérito Ψ para controlar o

tamanho dos passos. Esta função desempenha o papel da função objectivo em optimização sem restrições, uma vez que se exige uma redução significativa em Ψ , em cada iteração. Embora existam várias funções mérito que podiam ser usadas nos métodos PQS apenas se apresentam as que vão ser utilizadas.

Nesta secção, discutem-se as condições que devem ser impostas ao problema e às funções mérito, por forma a garantir um decréscimo nessas funções com base num passo gerado pelo método PQS.

A primeira função mérito a ser estudada é a penalidade l_2 que, para o problema com restrições de igualdade (5.1), é definida por

$$\Psi_2(x; \beta) = F(x) + \frac{\beta}{2} \|h(x)\|_2^2, \quad (5.16)$$

onde $\beta > 0$ denota o parâmetro de penalidade. O resultado seguinte descreve a derivada direcional de Ψ_2 ao longo da direcção Δx_k gerada pelo subproblema PQS. Por uma questão de simplificação de notação vai usar-se Δ_k em vez de Δx_k .

Lema 5.1 *Sejam Δ_k e λ_{k+1} os vectores gerados pela iteração PQS (5.9). Então, a derivada direcional de Ψ_2 na direcção de Δ_k satisfaz*

$$\nabla \Psi_2(x_k; \beta)^T \Delta_k \leq -\Delta_k^T W_k \Delta_k - (\beta \|h_k\|_2 - \|\lambda_{k+1}\|_2) \|h_k\|_2.$$

Prova. A função Ψ_2 é diferenciável e o gradiente no ponto (x_k, λ_k) é dado por

$$\nabla \Psi_2(x_k; \beta) = \nabla F_k + \beta \nabla h_k^T h_k.$$

Se Δ_k satisfaz as equações PQS (5.9), tem-se

$$\begin{aligned} \nabla \Psi_2(x_k; \beta)^T \Delta_k &= \nabla F_k^T \Delta_k + \beta h_k^T \nabla h_k \Delta_k \\ &= -\Delta_k^T W_k \Delta_k + \lambda_{k+1}^T \nabla h_k \Delta_k - \beta h_k^T h_k \\ &= -\Delta_k^T W_k \Delta_k - \lambda_{k+1}^T h_k - \beta \|h_k\|_2^2. \end{aligned}$$

Usando a desigualdade [6],

$$-\lambda_{k+1}^T h_k \leq \|\lambda_{k+1}\|_2 \|h_k\|_2$$

obtém-se

$$\nabla \Psi_2(x_k; \beta)^T \Delta_k \leq -\Delta_k^T W_k \Delta_k - (\beta \|h_k\|_2 - \|\lambda_{k+1}\|_2) \|h_k\|_2. \quad (5.17)$$

■

De (5.17) conclui-se que Δ_k é uma direcção descendente para Ψ_2 , se W_k for definida positiva e β suficientemente grande. Uma análise mais cuidada mostra que a hipótese sobre W_k pode ser relaxada, sendo apenas necessário que W_k seja definida positiva no

espaço tangente das restrições. Um valor apropriado para o parâmetro de penalidade deve satisfazer a condição

$$\beta > \frac{\|\lambda_{k+1}\|_2}{\|h_k\|_2}.$$

A outra função mérito a implementar é a função Lagrangeana aumentada de Fletcher (para um valor fixo de λ), que é dada por

$$\Psi_F(x, \lambda; \beta) = F(x) - \lambda^T h(x) + \frac{\beta}{2} \|h(x)\|_2^2, \quad (5.18)$$

onde $\beta > 0$ é o parâmetro de penalidade.

Lema 5.2 *Sejam Δ_k e λ_{k+1} os vectores gerados pela iteração PQS (5.9). Então, a derivada direccionada de Ψ_F na direcção de Δ_k satisfaz*

$$\nabla \Psi_F(x_k, \lambda_k; \beta)^T \Delta_k \leq -\Delta_k^T W_k \Delta_k - (\beta \|h_k\|_2 - \|\lambda_{k+1}\|_2 - \|\lambda_k\|_2) \|h_k\|_2.$$

Prova. A função Ψ_F é diferenciável e o gradiente no ponto (x_k, λ_k) é dado por

$$\nabla \Psi_F(x_k, \lambda_k; \beta) = \nabla F(x_k) - \nabla h^T(x_k) \lambda_k + \beta \nabla h^T(x_k) h(x_k)$$

Se Δ_k satisfaz as equações PQS (5.9), tem-se

$$\begin{aligned} \nabla \Psi_F(x_k, \lambda_k; \beta)^T \Delta_k &= \nabla F_k^T \Delta_k - \lambda_k^T \nabla h_k \Delta_k + \beta h_k^T \nabla h_k \Delta_k \\ &= -\Delta_k^T W_k \Delta_k + \lambda_{k+1}^T \nabla h_k \Delta_k + \lambda_k^T h_k - \beta h_k^T h_k \\ &= -\Delta_k^T W_k \Delta_k - \lambda_{k+1}^T h_k + \lambda_k^T h_k - \beta \|h_k\|_2^2. \end{aligned}$$

Usando as desigualdades [6]

$$-\lambda_{k+1}^T h_k \leq \|\lambda_{k+1}\|_2 \|h_k\|_2$$

e

$$\lambda_k^T h_k \leq \|\lambda_k\|_2 \|h_k\|_2$$

obtém-se

$$\nabla \Psi_F(x_k, \lambda_k; \beta)^T \Delta_k \leq -\Delta_k^T W_k \Delta_k - (\beta \|h_k\|_2 - \|\lambda_{k+1}\|_2 - \|\lambda_k\|_2) \|h_k\|_2.$$

■

Portanto, Δ_k é uma direcção descendente para Ψ_F , se W_k for definida positiva e se β satisfizer a condição:

$$\beta > \frac{\|\lambda_{k+1}\|_2 + \|\lambda_k\|_2}{\|h_k\|_2}.$$

Em ambos os casos, um valor apropriado para o parâmetro de penalidade, das funções Ψ_2 e Ψ_F , é obtido seleccionando uma constante $\bar{\delta} > 0$ e definindo β em cada iteração como sendo, respectivamente,

$$\beta_k = \frac{\|\lambda_{k+1}\|_2}{\|h_k\|_2} + \bar{\delta} \quad (5.19)$$

e

$$\beta_k = \frac{\|\lambda_{k+1}\|_2 + \|\lambda_k\|_2}{\|h_k\|_2} + \bar{\delta}. \quad (5.20)$$

O teorema seguinte sintetiza os resultados apresentados nesta secção, que englobam um procedimento de geração do passo e a escolha da função mérito nos métodos PQS.

Teorema 5.1 *Suponha que x_k não é um ponto estacionário do problema com restrições de igualdade (5.1) e que a Hessiana reduzida $Z_k^T W_k Z_k$ é definida positiva. Então, a direcção de procura Δ_k , gerada pela iteração PQS (5.9), é uma direcção descendente para as funções mérito Ψ_2 e Ψ_F , se β for dado respectivamente por (5.19) e (5.20).*

Na prática, é desejável que os valores da sucessão $\{\beta_k\}$ permaneçam inalterados sempre que possível, à medida que os iterandos convergem para a solução. Este objectivo é atingido, usando uma regra de actualização que não altere o valor β sempre que ele pareça adequado, e que o aumente de uma quantidade significativa, caso contrário. A correspondente regra consiste em definir

$$\beta_k = \begin{cases} \beta_{k-1} & \text{se } \beta_{k-1} \geq \bar{\gamma}_k + \delta \\ \bar{\gamma}_k + 2\delta & \text{caso contrário,} \end{cases} \quad (5.21)$$

para algum $\delta > 0$ ($\delta = 10^{-2}$, por exemplo), em que $\bar{\gamma}_k$ é dado por

$$\bar{\gamma}_k = \begin{cases} \frac{\|\lambda_{k+1}\|_2}{\|h_k\|_2}, & \text{no caso da função mérito } \Psi_2 \\ \frac{\|\lambda_{k+1}\|_2 + \|\lambda_k\|_2}{\|h_k\|_2}, & \text{no caso da função mérito } \Psi_F. \end{cases}$$

A regra (5.21) define uma sucessão de valores para β_k monótona crescente, mas isto nem sempre é desejável na prática. Em alguns casos, β_k , toma um valor muito grande

no início das iterações, penalizando bastante as restrições durante o resto do processo iterativo. Por conseguinte, algumas implementações PQS incluem heurísticas que permitem decrementar β_k em certas iterações, sem interferir com as propriedades de convergência global da iteração.

5.6 O efeito de Maratos

Algumas funções mérito podem impedir uma rápida convergência dos métodos PQS ao rejeitarem passos que originam um bom progresso em direcção a uma solução. Este fenómeno é muitas vezes designado por *efeito de Maratos*. Em [8] é apresentado um problema (Exemplo 18.1), no qual os passos PQS originam uma taxa de convergência quadrática mas causam um crescimento tanto no valor da função objectivo como na norma da restrição. Como consequência, estes passos serão rejeitados por muitas funções mérito.

Em [8], mostra-se que qualquer função mérito da forma

$$\Psi(x; \beta) = F(x) + \beta c(h(x)),$$

onde $c(\cdot)$ é uma função não negativa que satisfaz $c(0) = 0$, origina uma rejeição do passo Δx_k . Assim, qualquer algoritmo que se baseie numa função mérito deste tipo sofre do efeito de Maratos. Incluem-se neste tipo de função mérito a função de penalidade Ψ_2 e a função não diferenciável $F(x) + \beta \|h(x)\|_1$.

Se não forem tomadas precauções, o efeito de Maratos pode abrandar dramaticamente os métodos PQS. Lista-se, de seguida, algumas técnicas que evitam o efeito de Maratos.

- (a) Uso de uma função mérito que não sofra do efeito de Maratos. Um exemplo é a função Lagrangeana aumentada de Fletcher (5.18), para a qual se pode mostrar que os passos gerados por (5.9) são aceites numa vizinhança da solução que satisfaça as condições suficientes de segunda ordem.
- (b) Uso de uma correcção de segunda ordem, que consiste em adicionar a Δx_k um passo Δc_k que é calculado em $h(x + \Delta x_k)$ e que garante uma redução significativa no valor das restrições [8].
- (c) Uso de uma condição que permite que a função Ψ incremente em certas iterações, que é equivalente a uma estratégia não monótona. Um exemplo é a técnica *watchdog* apresentada em [2] e [8].

No algoritmo PQS implementado nesta tese, foram usadas as duas primeiras técnicas.

Quando se usa a função mérito de penalidade l_2 , recorre-se à técnica de correcção de segunda ordem para evitar o efeito de Maratos.

Adicionando um termo de correcção que origina uma maior redução nas restrições, a iteração PQS ultrapassa as dificuldades associadas ao efeito de Maratos.

Dado um passo Δx_k PQS, o passo de correcção Δc_k é definido como sendo

$$\Delta c_k = -\nabla h_k^T (\nabla h_k \nabla h_k^T)^{-1} h(x_k + \Delta x_k).$$

O passo Δc_k satisfaz uma linearização das restrições h no ponto $x_k + \Delta x_k$, isto é,

$$\nabla h_k \Delta c_k + h(x_k + \Delta x_k) = 0.$$

De facto, Δc_k corresponde à solução do mínimo da norma desta equação. O efeito do passo de correcção Δc_k , que é normal às restrições, consiste em decrementar $\|h(x)\|$ da ordem de $\|x_k - x^*\|^3$. Assim, garante-se que o passo que vai de x_k para $x_k + \Delta x_k + \Delta c_k$ decrementa a função mérito, pelo menos, numa vizinhança da solução. O preço a pagar é o cálculo adicional da função de restrição h em $x_k + \Delta x_k$.

Algoritmo PQS

Dado um ponto inicial (x_1, λ_1) , $\beta_1=1$ e uma matriz B_1 $n \times n$ simétrica e definida positiva, para $k = 1, 2, \dots$

1. Resolver o sistema (5.8) com W_k substituído por B_k , para obter Δ_k e π_k .
2. Fazer $\lambda_{k+1} = \pi_k$
3. Determinar β_{k+1} pela regra (5.21).
4. Calcular o comprimento do passo, α_k , através de um algoritmo de procura unidimensional, nomeadamente através do critério de Armijo, que garanta uma redução significativa em Ψ_F [no caso da função Ψ_2 , usar o Algoritmo CSO (correcção de segunda ordem) para calcular Δc_k].
5. Fazer $x_{k+1} = x_k + \alpha_k \Delta_k$ [ou $x_{k+1} = x_k + \Delta_k + \Delta c_k$].
6. Se o novo ponto satisfaz o critério de paragem então parar.
7. Actualizar B_{k+1} pela fórmula de actualização BFGS.

Algoritmo CSO (Correcção de segunda ordem)

Fazer $\alpha_k = 1$

Enquanto $\left(\Psi_2(x_k + \alpha_k \Delta_k; \beta_k) > \Psi_2(x_k; \beta_k) + 0.1 \alpha_k \nabla \Psi_2(x_k; \beta_k)^T \Delta_k \right)$ fazer

Se $\alpha_k = 1$ então

calcular $\Delta c_k = -\nabla h_k^T (\nabla h_k \nabla h_k^T)^{-1} h(x_k + \Delta_k)$;

Se $\Psi_2(x_k + \Delta_k + \Delta c_k; \beta_k) \leq \Psi_2(x_k; \beta_k) + 0.1 \nabla \Psi_2(x_k; \beta_k)^T \Delta_k$ então

Terminar e executar a segunda parte do Passo 5 do Algoritmo PQS

FimSe

```

FimSe
 $\alpha_k \leftarrow \alpha_k/2$ 
Se  $\alpha_k \|\Delta_k\|_2 \leq 10^{-8}$  então
     $\alpha_k \leftarrow 1$  e Terminar
FimSe
FimEnquanto
Executar a primeira parte do Passo 5 do Algoritmo PQS.

```

Critério de Paragem

O processo iterativo termina se

- se $\|\nabla_x \mathbb{L}\|_\infty \leq \text{máx}(\epsilon_1, \text{'zero'})$ e $\|h\|_\infty \leq \text{máx}(\epsilon_2, \text{'zero'})$; ou
- se o número de iterações exced o limite máximo pré-estabelecido (itMax)

onde $\|\cdot\|_\infty$ representa a norma infinito e 'zero' denota a precisão da máquina.

5.7 Exercícios

1. Usando o método PQS, resolva o seguinte problema

$$\underset{x \in \mathbb{R}^3}{\text{minimizar}} F(x) \equiv 4x_1^2 + 2x_2^2 + 2x_3^2 - 33x_1 + 16x_2 - 24x_3$$

sujeito a

$$\begin{aligned} 3x_1 - 2x_2^2 &= 7 \\ 4x_1 + x_3^2 &= 11. \end{aligned}$$

O processo iterativo deve ser iniciado com o ponto inicial $\tilde{x}_1 = (4, -3, 4)$ e $\tilde{\lambda}_1 = (1, 1)$. Implemente duas iterações usando o método dual na resolução do sistema KKT e use $H_0 = I$ como aproximação inicial à inversa da Hessiana. Deve, também, implementar o algoritmo baseado no critério de Armijo com a função mérito Ψ_F para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.005$.

2. Considere o seguinte problema

$$\underset{x \in \mathbb{R}^3}{\text{minimizar}} F(x) \equiv (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4$$

sujeito a

$$x_1(1 + x_2^2) + x_3^4 = 4 + 3\sqrt{2}$$

Implemente duas iterações do método PQS. O processo iterativo deve ser iniciado com o ponto inicial $\tilde{x}_1 = (1.5, 1.5, 1.5)$ e $\tilde{\lambda}_1 = 1$. Use o método dual na resolução do sistema KKT e use $H_0 = I$ como aproximação inicial à inversa da Hessiana. Deve, também, implementar o algoritmo baseado no critério de Armijo com a função mérito Ψ_2 para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.05$.

3. Considere o seguinte problema

$$\underset{x \in \mathbb{R}^4}{\text{minimizar}} F(x) \equiv (x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$$

sujeito a

$$\begin{aligned} x_1 - 2 &= 0 \\ x_3^2 + x_4^2 - 2 &= 0 \end{aligned}$$

Implemente duas iterações do método PQS. O processo iterativo deve ser iniciado com o ponto inicial $\tilde{x}_1 = (1, 1, 1, 1)$ e $\tilde{\lambda}_1 = (0, 0)$. Use o método dual na resolução do sistema KKT e use $H_0 = I$ como aproximação inicial à inversa da Hessiana. Deve, também, implementar o algoritmo baseado no critério de Armijo com a função mérito Ψ_F para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.05$.

4. Considere o seguinte problema

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad F(x) \equiv x_1^2 + 4x_2^2$$

sujeito a

$$x_1 + 2x_2 = 1$$

Implemente duas iterações do método PQS. O processo iterativo deve ser iniciado com o ponto inicial $\tilde{x}_1 = (1, 1)$ e $\tilde{\lambda}_1 = 1$. Use o método dual na resolução do sistema KKT e use $H_0 = I$ como aproximação inicial à inversa da Hessiana. Deve, também, implementar o algoritmo baseado no critério de Armijo com a função mérito Ψ_F para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.08$.

5. Considere o seguinte problema

$$\underset{x \in \mathbb{R}^2}{\text{minimizar}} \quad F(x) \equiv (1 - x_1)^2$$

sujeito a

$$10(x_2 - x_1^2) = 0$$

Implemente duas iterações do método PQS. O processo iterativo deve ser iniciado com o ponto inicial $\tilde{x}_1 = (-1.2, 1)$ e $\tilde{\lambda}_1 = 1$. Use o método dual na resolução do sistema KKT e use $H_0 = I$ como aproximação inicial à inversa da Hessiana. Deve, também, implementar o algoritmo baseado no critério de Armijo com a função mérito Ψ_F para calcular o comprimento do passo, α , em cada iteração. Considere $\mu_1 = 0.08$.

6. Implemente em MatLab o método da PQS para resolver problemas de otimização não linear com restrições de igualdade. Use $H_0 = I$ como aproximação inicial à inversa da Hessiana e inclua o seguinte:

- Resolva o sistema KKT pelo método dual.
- Use a estratégia de redução sucessiva do passo apresentada na subsecção 2.3.1 com a função mérito Ψ_F e Ψ_2 . Considere $\mu_1 = 0.1$, $\beta_1 = 1$ e $\delta = 10^{-2}$.
- Utilize a estratégia de *skipping* apresentada no Capítulo 2 com parâmetro de recomeço $\gamma = 10^{-12}$.
- Aceite (x^*, λ^*) como uma solução do problema de otimização se $\|\nabla_x \mathbb{L}\|_\infty \leq \epsilon_1$ e $\|h\|_\infty \leq \epsilon_2$, ou se o número de iterações excede o limite máximo pré-estabelecido (itMax). Use $\epsilon_1 = \epsilon_2 = 10^{-6}$ e itMax=100.
- Escreva o valor inicial, a nova aproximação à solução bem como as normas do vector gradiente da função Lagrangeano e do vector das funções de restrição, no ponto óptimo. Indique também se não foi encontrada nenhuma solução ao fim do limite máximo de iterações pré-estabelecido.

(f) Teste os algoritmos com os seguintes problemas teste:

• **HS6**

$$\min_{x \in \mathbb{R}^2} F(x) \equiv (1 - x_1)^2$$

$$\text{s.a. } 10(x_2 - x_1^2) = 0$$

Considere $x_0 = (-1.2, 1)$. Solução: $x^* = (1, 1)$; $F(x^*) = 0$.

• **HS7**

$$\min_{x \in \mathbb{R}^2} F(x) \equiv \ln(1 + x_1^2) - x_2$$

$$\text{s.a. } (1 + x_1^2)^2 + x_2^2 - 4 = 0$$

Considere $x_0 = (2, 2)$. Solução: $x^* = (0, \sqrt{3})$; $F(x^*) = -\sqrt{3}$.

• **HS8**

$$\min_{x \in \mathbb{R}^2} F(x) \equiv -1$$

$$\text{s.a. } x_1^2 + x_2^2 - 25 = 0$$

$$x_1 x_2 - 9 = 0$$

Considere $x_0 = (2, 1)$. Solução: $x^* = (a, \frac{9}{a}), (-a, -\frac{9}{a}), (b, \frac{9}{b}), (-b, -\frac{9}{b})$;

onde $a = \sqrt{\frac{25+\sqrt{301}}{2}}$ e $b = \sqrt{\frac{25-\sqrt{301}}{2}}$; $F(x^*) = -1$.

• **HS9**

$$\min_{x \in \mathbb{R}^2} F(x) \equiv \sin\left(\frac{\pi x_1}{12}\right) \cos\left(\frac{\pi x_2}{16}\right)$$

$$\text{s.a. } 4x_1 - 3x_2 = 0$$

Considere $x_0 = (0, 0)$. Solução: $x^* = (12k-3, 16k-4)$; $k = 0, \pm 1, \pm 2, \dots$ $F(x^*) = -0.5$.

• **HS26**

$$\min_{x \in \mathbb{R}^3} F(x) \equiv (x_1 - x_2)^2 + (x_2 - x_3)^4$$

$$\text{s.a. } (1 + x_2^2)x_1 + x_3^4 - 3 = 0$$

Considere $x_0 = (-2.6, 2, 2)$. Solução: $x^* = (1, 1, 1)$; $F(x^*) = 0$.

• **HS27**

$$\min_{x \in \mathbb{R}^3} F(x) \equiv 0.01(x_1 - 1)^2 + (x_2 - x_1^2)^2$$

$$\text{s.a. } x_1 + x_3^2 + 1 = 0$$

Considere $x_0 = (2, 2, 2)$. Solução: $x^* = (-1, 1, 0)$; $F(x^*) = 0.04$.

• **HS28**

$$\min_{x \in \mathbb{R}^3} F(x) \equiv (x_1 + x_2)^2 + (x_2 + x_3)^2$$

$$\text{s.a. } x_1 + 2x_2 + 3x_3 - 1 = 0$$

Considere $x_0 = (-4, 1, 1)$. Solução: $x^* = (0.5, -0.5, 0.5)$; $F(x^*) = 0$.

• **HS39**

$$\min_{x \in \mathbb{R}^4} F(x) \equiv -x_1$$

$$\text{s.a. } x_2 - x_1^3 - x_3^2 = 0$$

$$x_1^2 - x_2 - x_4^2 = 0$$

Considere $x_0 = (2, 2, 2, 2)$. Solução: $x^* = (1, 1, 0, 0)$; $F(x^*) = -1$.

• **HS40**

$$\min_{x \in \mathbb{R}^4} F(x) \equiv -x_1 x_2 x_3 x_4$$

$$\text{s.a. } x_1^3 + x_2^2 - 1 = 0$$

$$x_1^2 x_4 - x_3 = 0$$

$$x_4^2 - x_2 = 0$$

Considere $x_0 = (0.8, 0.8, 0.8, 0.8)$. Solução: $x^* = (2^a, 2^{2b}, (-1)^i 2^c, (-1)^i 2^b)$; $i = 1, 2$; $a = -\frac{1}{3}$, $b = -\frac{1}{4}$, $c = -\frac{11}{12}$, $F(x^*) = -0.25$.

• **HS42**

$$\min_{x \in \mathbb{R}^4} F(x) \equiv (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$$

$$\text{s.a. } x_1 - 2 = 0$$

$$x_3^2 + x_4^2 - 2 = 0$$

Considere $x_0 = (1, 1, 1, 1)$. Solução: $x^* = (2, 2, 0.6\sqrt{2}, 0.8\sqrt{2})$; $F(x^*) = 28 - 10\sqrt{2}$.

• **HS46**

$$\min_{x \in \mathbb{R}^5} F(x) \equiv (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$$

$$\begin{aligned} \text{s.a. } x_1^2 x_4 + \sin(x_4 - x_5) - 1 &= 0 \\ x_2 + x_3^4 x_4^2 - 2 &= 0 \end{aligned}$$

Considere $x_0 = (0.5\sqrt{2}, 1.75, 0.5, 2, 2)$. Solução: $x^* = (1, 1, 1, 1, 1)$; $F(x^*) = 0$.

• **HS47**

$$\min_{x \in \mathbb{R}^5} F(x) \equiv (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4$$

$$\begin{aligned} \text{s.a. } x_1 + x_2^2 + x_3^3 - 3 &= 0 \\ x_2 - x_3^2 + x_4 - 1 &= 0 \\ x_1 x_5 - 1 &= 0 \end{aligned}$$

Considere $x_0 = (2, \sqrt{2}, -1, 2 - \sqrt{2}, 0.5)$. Solução: $x^* = (1, 1, 1, 1, 1)$; $F(x^*) = 0$.

• **HS48**

$$\min_{x \in \mathbb{R}^5} F(x) \equiv (x_1 - 1)^2 + (x_2 - x_3)^2 + (x_4 - x_5)^2$$

$$\begin{aligned} \text{s.a. } x_1 + x_2 + x_3 + x_4 + x_5 - 5 &= 0 \\ x_3 - 2(x_4 + x_5) + 3 &= 0 \end{aligned}$$

Considere $x_0 = (3, 5, -3, 2, -2)$. Solução: $x^* = (1, 1, 1, 1, 1)$; $F(x^*) = 0$.

• **HS49**

$$\min_{x \in \mathbb{R}^5} F(x) \equiv (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$$

$$\begin{aligned} \text{s.a. } x_1 + x_2 + x_3 + 4x_4 - 7 &= 0 \\ x_3 + 5x_5 - 6 &= 0 \end{aligned}$$

Considere $x_0 = (10, 7, 2, -3, 0.8)$. Solução: $x^* = (1, 1, 1, 1, 1)$; $F(x^*) = 0$.

• HS50

$$\min_{x \in \mathbb{R}^5} F(x) \equiv (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^2$$

$$\text{s.a. } x_1 + 2x_2 + 3x_3 - 6 = 0$$

$$x_2 + 2x_3 + 3x_4 - 6 = 0$$

$$x_3 + 2x_4 + 3x_5 - 6 = 0$$

Considere $x_0 = (35, -31, 11, 5, -5)$. Solução: $x^* = (1, 1, 1, 1, 1)$; $F(x^*) = 0$.

Bibliografia

- [1] M. Bartholomew-Biggs. The estimation of the Hessian in nonlinear least squares problems with non-zero residuals. *Mathematical Programming*, 12:67–80, 1977.
- [2] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, and H. C. Pederson. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming*, 16:1–17, 1982.
- [3] J. E. Dennis, D. M. Gay, and R. E. Welsch. An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, 7(3):348–368, 1981.
- [4] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [5] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, 2 edition, 1987.
- [6] H. Lutkepohl. *Handbook of Matrices*. John Wiley and Sons, 1996.
- [7] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill Series in Industrial Engineering, Management Science, 1996.
- [8] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 1999.
- [9] M. A. Wolfe. *Numerical Methods for Unconstrained Optimization*. Van Nostrand Reinhold Company, 1978.
- [10] H. Yabe. Variations of structured Broyden families for nonlinear least squares problems. *Optimization Methods and Software*, 2:107–144, 1993.
- [11] H. Yabe and T. Takahashi. Factorized quasi-Newton methods for nonlinear least squares problems. *Mathematical Programming*, 51:75–100, 1991.