Escola de Engenharia da Universidade do Minho
Mestrado integrado em Engenharia de Telecomunicações e Informática

TRABALHO PRÁTICO 2
Parte A

Métodos numéricos e Otimização não linear

Bruno Xavier Brás dos Santos, nºa72122

Rui Filipe Ribeiro Freitas, nº a84121

Tiago João Pereira Ferreira, nº a85392

03 de junho de 2020

Parte A

$$\begin{array}{r} 72122 \\ 84121 \\ +\ 85392 \\ \hline 241635 \end{array} \rightarrow \text{ÍMPAR}$$

1

$$f(u_1,u_2) = \frac{1}{3}u_1^3 + \frac{1}{2}u_1^2 + 2u_1u_2 + \frac{1}{2}u_2^2 - u_2 + 1$$

$$\nabla f(u_1,u_2) = \begin{pmatrix} u_1^2 + u_1 + 2u_2 \\ 2u_1 + u_2 - 1 \end{pmatrix}$$

$$\nabla f(u_1,u_2) = 0 \Leftrightarrow \begin{cases} u_1^2 + u_1 + 2u_2 = 0 \\ 2u_1 + u_2 - 1 = 0 \end{cases} \Leftrightarrow \begin{cases} u_1^2 + u_1 + 2(1-2u_1) = 0 \\ u_2 = 1 - 2u_1 \end{cases} \Leftrightarrow$$

$$\begin{cases} u_1^2 + u_1 + 2 - 4u_1 = 0 \\ \_\_\_\_ \end{cases} \Leftrightarrow \begin{cases} u_1^2 - 3u_1 + 2 = 0 \\ \_\_\_\_ \end{cases} \Leftrightarrow$$

$$\begin{cases} u_1 = 1 \lor u_1 = 2 \\ u_2 = 1-2 \lor u_2 = 1-4 \end{cases} \Leftrightarrow \begin{cases} u_1 = 1 \lor u_1 = 2 \\ u_2 = -1 \lor u_2 = -3 \end{cases}$$

$$\nabla^2 f(u_1,u_2) = \begin{pmatrix} 2u_1+1 & 2 \\ 2 & 1 \end{pmatrix}$$

Para $(1,-1)$

$$\nabla^2 f(1,-1) = \begin{array}{cc} da & db \\ \end{array} \begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix} \qquad \begin{array}{l} da = 3 > 0 \\ db = -1 < 0 \end{array}$$

Matriz indefinida, o ponto $(1,-1)$ é um ponto sela.

Para $(2,-3)$

$$\nabla^2 f(2,-3) = \begin{array}{cc} da & db \\ \end{array} \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix} \qquad \begin{array}{l} da = 5 > 0 \\ db = 1 > 0 \end{array}$$

Matriz definida positiva, o ponto $(2,-3)$ é um ponto minimizante

(2)

$t_1 = 7$
$\delta = 1$
$M = 0{,}5$

$h(t) = -0{,}125t^4 + 2{,}5t^3 - 12{,}9t^2 - 1{,}1t + 94{,}8$

$t \to u$

$h(t) \to f(u)$

$f(u) = -0{,}125u^4 + 2{,}5u^3 - 12{,}9u^2 - 1{,}1u + 94{,}8$

1ª iteração

$\begin{cases} u_1 = 7 \\ f(u_1) = 12{,}375 \end{cases}$ procurar para a direita

$\begin{cases} u_2 = 7 + \delta = 8 \\ f(u_2) = 28{,}4 \end{cases}$ aumentou, logo procurar para a esquerda

$\begin{cases} u_{-1} = 7 - \delta = 6 \\ f(u_{-1}) = 1{,}8 \end{cases}$ continuar

$\begin{cases} u_{-2} = 6 - 2\delta = 4 \\ f(u_{-2}) = 12 \end{cases}$ parar e calcular ponto médio

$\begin{cases} u_m = \frac{u_{-2} + u_{-1}}{2} = 5 \\ f(u_m) = 1{,}175 \end{cases}$

$f(u_{-1}) > f(u_m) \Rightarrow$ Escolher os pontos u_{-2}, u_m e u_{-1}

$u_1 \leftarrow 4$ $\quad f(u_1) = 12$
$u_2 \leftarrow 5$ $\quad f(u_2) = 1{,}175$ $\quad \Delta = 1$
$u_3 \leftarrow 6$ $\quad f(u_3) = 1{,}8$

$$u^*(q) = u_2 + \frac{\Delta\, f(u_1) - f(u_3)}{2(f(u_3) - 2f(u_2) + f(u_1))} = 5 + \frac{12 - 1{,}8}{2(1{,}8 - 2 \times 1{,}175 + 12)} = 5{,}4454$$

C.P

$\delta = M\delta = 1 \times 0{,}5 = 0{,}5$

2ª. iterasão

$$\begin{cases} u_1 = u^*(q) = 5{,}4454 \\ f(u_1) = 0{,}0591 \end{cases}$$ procurar para a direita

$$\begin{cases} u_2 = u_1 + \delta = 5{,}9454 \\ f(u_2) = 1{,}4822 \end{cases}$$ aumentou, procurar para a esquerda

$$\begin{cases} u_{-1} = u_1 - \delta = 4{,}9454 \\ f(u_{-1}) = 1{,}4709 \end{cases}$$ parar e calcular ponto médio

$$\begin{cases} um = u_1 = 5{,}4454 \\ f(um) = f(u_1) = 0{,}0591 \end{cases}$$

$$\boxed{\text{Escolher os pontos } u_{-1}, um \text{ e } u_2}$$

$u_1 \leftarrow 4{,}9454$ $\quad f(u_1) = 1{,}4709$ $\quad \Delta = 0{,}5$

$v_2 \leftarrow 5{,}4454$ $\quad f(v_2) = 0{,}0591$

$u_3 \leftarrow 5{,}9454$ $\quad f(v_3) = 1{,}4822$

$$u^*(q) = u_2 + \Delta \frac{f(u_1) - f(v_3)}{2(f(v_3) - 2f(u_2) + f(u_1))} =$$

$$= 5{,}4454 + 0{,}5 \times \frac{1{,}4709 - 1{,}4822}{2 \times (1{,}4822 - 2 \times 0{,}0591 + 1{,}4709)}$$

$$= 5{,}4444$$

$$u^*(q) = 5{,}4444$$

$$f(u^*(q)) = 0{,}0590$$

Instante do tempo $\quad t = 5{,}4444$

Distância da água ao fundo do vale $h(t) = 0{,}0590$

Escola de Engenharia da Universidade do Minho
Mestrado integrado em Engenharia de Telecomunicações e Informática

TRABALHO PRÁTICO 2
Parte B

Métodos numéricos e Otimização não linear

Bruno Xavier Brás dos Santos, nºa72122

Rui Filipe Ribeiro Freitas, nº a84121

Tiago João Pereira Ferreira, nº a85392

03 de junho de 2020

i) Descrição, enquadramento do problema e respetiva formulação matemática

No âmbito da unidade curricular de métodos numéricos e otimização não linear ministrada no curso de MIETI na universidade do Minho foi proposto um segundo projeto que consistia na procura de um problema de otimização não linear sem restrições nas variáveis cuja resolução deve ser realizada utilizando a rotina fminunc do MATLAB.

O problema encontrado foi retirado de um conjunto de slides, página 7, denominado "Business Optimization and Simulation - Module 4 - Nonlinear optimization" que pertencem à Universidad Carlos III de Madrid

O problema supramencionado tem como título "Smartphone Marketing" e é descrito como:

" Uma empresa deseja vender um smartphone para competir com outros produtos de última geração. A empresa investiu cem mil euros para desenvolver o produto. O sucesso do produto dependerá do valor investido na campanha de marketing e o preço final do smartphone. As duas principais decisões, ou seja, as variáveis independentes a calcular são:

- a: valor para investir na campanha de marketing
- p: preço do smartphone

A fórmula usada pelo departamento de marketing para estimar as vendas do novo produto durante o próximo ano é:

$$S = 20000 + 5\sqrt{a} - 60p$$

O custo de produção do smartphone é de 100 euros/unidade. Como a empresa poderia maximizar seus lucros para os próximos anos? "

Formulação do problema:

- Lucros das vendas: $\left(20000 + 5\sqrt{a} - 60p\right)p$
- Custos totais produção $\left(20000 + 5\sqrt{a} - 60p\right)100$
- Custos de desenvolvimento: 100000
- Custos de marketing: a
- Preço de venda do smartphone: p
- Lucro total: $\left(20000 + 5\sqrt{a} - 60p\right)(p - 100) - 100000 - a$

O objetivo é maximizar o lucro total.

ii) Objetivo e condições de aplicabilidade da função fminunc

O objetivo é encontrar um problema de otimização não linear sem restrições nas variáveis cuja resolução deve ser realizada utilizando a rotina fminunc do MATLAB.

A rotina fminunc implementa um método do gradiente para encontrar o mínimo da função passada como parâmetro. Só pode ser aplicada em funções diferenciáveis, uma vez que para além do valor da função, também usa o valor do gradiente e da matriz hessiana.

O problema escolhido pelo grupo tem como objetivo maximizar o lucro total. Uma vez que a rotina fminunc implementa um algoritmo para minimizar, usaremos a seguinte mudança na função objetivo : - f(- x). Deste modo, minimizaremos a função - f(- x). Na solução final, o minimizante da respetiva função corresponde ao maximizante da função objetivo e o lucro total é o simétrico do lucro devolvido pela rotina fminunc.

A função objetivo é a seguinte:

$$\max_{(a,p)\in\mathbb{R}^2} f(a,p) = \left(20000 + 5\sqrt{a} - 60p\right)(p - 100) - 100000 - a$$

A função objetivo é diferenciável, deste modo, é possível aplicar a rotina fminunc.

A sintaxe utilizada da rotina fminunc utilizada na resolução do presente problema foi:

```
[xmin, fmin, exitflag, output]=fminunc('smartphone', [1,1],options)
```

Como é possível observar, esta rotina recebe como parâmetros o ficheiro onde a função a minimizar foi implementada (smartphone.m), um vetor com a aproximação inicial e, opcionalmente, um conjunto de opções. A rotina devolve um vetor com o valor da solução (minimizantes) - xmin, o mínimo – fmin, o valor da exitflag e uma estrutura com dados importantes – output.

iii) Ficheiro .m, com a implementação do problema

Na imagem seguinte encontra-se o ficheiro com extensão '.m', onde é possível ver a função colocada no primeiro parâmetro da rotina fminunc. Na primeira linha encontra-se o nome da função – smartphone. Na terceira linha encontra-se a função objetivo. Como o objetivo é maximizar a função, e o algoritmo presente na rotina apenas encontra o minimizante da função, foi necessário fazer f=-S .

```
1     function [f] = smartphone(x)
2     %   Detailed explanation goes here
3 -   S=(20000 + 5*sqrt(x(1)) - 60*x(2))*(x(2)-100)-100000-x(1);
4 -   f=-S;
5 -   end
```

Para os vários testes que efetuamos com distintos valores iniciais, elaboramos o script que encontra se na figura seguinte.

```
1 -  close all; clear; clc;
2
3 -  [xmin, fmin, exitflag, output]=fminunc('smartphone', [1,1]);
4
5 -  disp('Resultado da rotina: ');
6 -  disp('Valor dos maximizantes: ');            disp(xmin);
7 -  disp('Valor do máximo: ');                   disp(-fmin);
8 -  disp('Valor da EXITFLAG: ');                 disp(exitflag);
9 -  disp('Estrutura OUTPUT: ');                  disp(output);
```

Na linha 3 do script encontra-se a execução da rotina. As linhas 5 até ao final têm como finalidade a exibição dos resultados da execução da rotina

iv) Análise de resultados

Na figura seguinte é possível ver os resultados devolvidos através da execução da rotina:

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
Resultado da rotina:
Valor dos maximizantes:
   1.0e+05 *

    1.0599    0.0023

Valor do máximo:
   8.1163e+05

Valor da EXITFLAG:
     1

Estrutura OUTPUT:
       iterations: 22
        funcCount: 75
         stepsize: 380.9445
     lssteplength: 1
    firstorderopt: 0.0239
        algorithm: 'quasi-newton'
          message: '↵Local minimum found.↵↵Optimization completed because the s
```

O mínimo local foi encontrado. Sendo o seu valor (1.0599*10^5 ; 0.0023*10^5). Deste modo, a solução do problema é: (105990 ; 230) -> a = 105990 e p = 230.

O valor do máximo corresponde a f(a,p) = f(105990, 230) = 8.1163e+05 = 811630

No contexto do problema, o lucro máximo que a empresa pode conseguir na produção do smartphone é de 811 630 euros para um gasto de 105 990 euros com a campanha de marketing na publicidade do smartphone, sendo o preço de venda final do smartphone de 230 euros.

Através do valor da flag "exitflag = 1" podemos observar que o problema convergiu.

Analisando a estrutura "Output" concluímos que, para a aproximação inicial de [1, 1], a rotina realizou 22 iterações, foram realizados 75 cálculos da função objetivo e o algoritmo usado foi o "quasi-newton". A mensagem obtida da presente estrutura mostra-nos que o mínimo local foi encontrado.

O critério de primeira ordem foi atingido: 9.241625e-07, aproximadamente zero. Este valor é menor que options.OptimalityTolerance = 1.000000e-06. O valor de options.OptimalityTolerance pode ser ajustado.

Foram realizados vários cálculos da rotina para valores diferentes da aproximação inicial, mas os resultados foram sempre os mesmos, o que podemos concluir que a solução encontrada é única, ou seja, apenas existe um ponto estacionário e a matriz hessiana da função simétrica da função objetivo no único ponto estacionário é definida positiva.