

TEMA 6

AJUSTE DE DADOS EXPERIMENTAIS: O MÉTODO DOS MÍNIMOS QUADRADOS

Em muitas ocasiões possuem-se dados experimentais sabendo-se que esses dados traduzem um fenómeno físico em que algumas grandezas se relacionam de modo linear. Por exemplo, se se medir a temperatura num furo, a diferentes profundidades, haverá, em geral, um aumento de temperatura com a profundidade uma vez que nos aproximamos das fontes internas de calor. Sabemos que o aumento de temperatura está relacionado com o gradiente de temperatura dT/dz (gradiente geotérmico) e portanto, a temperatura à profundidade z_1 está relacionada com a temperatura à superfície (T_o) e o gradiente da temperatura do seguinte modo,

$$T(z_1) = T_o + \frac{dT}{dz} z_1 \quad (1)$$

Ora, esta equação tem a forma da equação da recta: $y = a + b z$, onde a é a temperatura à superfície e b o declive da recta. Podemos, pois, usar as temperaturas medidas para determinar o gradiente da temperatura. Se se possuírem apenas dois valores de temperatura, por exemplo à superfície (T_o) e à profundidade z_1 (seja T_1) o cálculo do gradiente é simples e será igual a $(T_1 - T_o)/z_1$. Mas como se calculará o valor do gradiente se se possuírem valores

da temperatura a N ($N > 2$) diferentes profundidades? Considerem-se os dados da Tabela 1.

Tabela 1. Temperaturas em profundidade

Z (m)	T (°C)
10	19.20
20	21.05
40	21.25
60	21.74
80	22.25
100	22.77
150	23.98

Os dados da tabela estão representados no gráfico da Figura 1. Pode observar-se que a globalidade dos dados não se “alinha” segundo uma linha recta. Os dois primeiros valores “estragam” esse comportamento. Estes valores estão claramente “contaminados” e não representam o comportamento geral da temperatura devida ao gradiente geotérmico e, portanto, não devem ser usados no cálculo do gradiente.

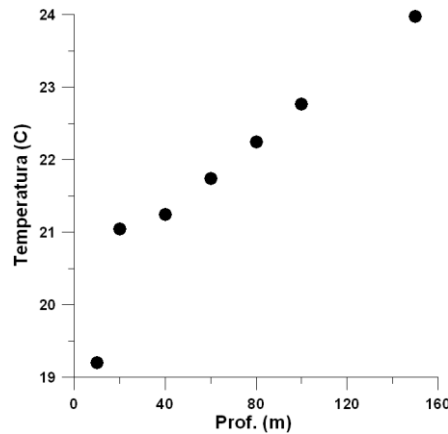


Figura 1. Temperatura versus profundidade (dados da Tabela 1).

Têm-se, então, cinco valores de temperatura (N=5) que podem ser usados no cálculo do gradiente e da temperatura à superfície. Em face destes dados poderia haver a tentação de determinarem-se 4 valores para o gradiente, usando-se as temperaturas a duas profundidades, fazendo-se em seguida a média desses valores. Por exemplo, usando-se os pares de temperaturas para as profundidades de 40 e 60m, 60 e 80 m, 80 e 100 m e de 100 e 150 m, obter-se-iam os seguintes valores para o gradiente : 0.0245, 0.0255, 0.0260, 0.0242 °C/m. Estes valores são muito próximos. A média destes valores será 0.0250 °C/m e o máximo desvio é 0.001 °C/m. Este resultado é aceitável (mas não o método)

Contudo, este não é o processo que deve ser seguido. O que se deve fazer é usar os cinco valores da temperatura e profundidade para se calcularem os valores do gradiente e da temperatura. Isto pode ser realizado do seguinte modo.

Considerando que a temperatura e a profundidade se relacionam de acordo com um relação linear da forma $T(z) = T_o + b z$ (equação da recta), tem-se que, de acordo com este modelo, os N valores da temperatura medido devem aproximar-se dos calculados com as seguintes equações:

$$\begin{aligned} T_1^c &= a + b z_1 \\ T_2^c &= a + b z_2 \\ &\dots \dots \dots \\ T_N^c &= a + b z_N \end{aligned} \tag{2}$$

Pretende-se, então, calcular os valores de b (gradiente geotérmico) e de a (temperatura à superfície) que satisfaçam os dados de acordo com um critério estabelecido. Esse critério pode ser por exemplo o seguinte: a diferença entre os valores estimados pelo modelo e os valores medidos deve ser mínima. Esta condição pode ser traduzida pela minimização de uma função Q, conhecida como função objectivo, definida por,

$$Q = \sum_{i=1}^N (T_i^o - T_i^c)^2 = \sum_{i=1}^N (T_i^o - a - b z_i)^2 \tag{3}$$

A minimização de Q é conseguida fazendo-se,

$$\frac{dQ}{da} = 0$$

$$\frac{dQ}{db} = 0$$

Vem,

$$2 \sum_{i=1}^N (T_i^o - a - b z_i) (-1) = 0$$

$$2 \sum_{i=1}^N (T_i^o - a - b z_i) (-z_i) = 0$$

Ou

$$-\sum_{i=1}^N T_i^o + a \sum_{i=1}^N 1 + b \sum_{i=1}^N z_i = 0$$

$$-\sum_{i=1}^N T_i^o z_i + a \sum_{i=1}^N z_i + b \sum_{i=1}^N z_i^2 = 0$$

Tem-se então o sistema de equações,

$$\begin{bmatrix} N & \sum_i^N z_i \\ \sum_i^N z_i & \sum_i^N z_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_i^N T_i^o \\ \sum_i^N T_i^o z_i \end{bmatrix} \quad (4)$$

A resolução deste sistema conduz a,

$$a = \frac{\sum z_i^2 \sum T_i^o - \sum z_i \sum z_i T_i^o}{\Delta}$$

$$b = \frac{N \sum z_i T_i^o - \sum z_i \sum T_i^o}{\Delta} \quad (5)$$

Com

$$\Delta = N \sum z_i^2 - \left(\sum z_i \right)^2$$

Para o exemplo da Tabela 1 ter-se-á,

$$\begin{bmatrix} 7 & 460 \\ 460 & 44600 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 152.24 \\ 10421.40 \end{bmatrix}$$

E resolução deste sistema (que pode ser feita manualmente) dá para a e b os valores, 19.84 e 0.029, respectivamente, isto é, a recta que se ajusta aos dados no sentido dos mínimos quadrados tem a equação:

$$T = 0.029 * z + 19.84$$

Um cálculo com maior precisão conduz à equação:

$$T = 0.0249 * z + 20.257$$

Considere-se de novo o problema do cálculo do gradiente geotérmico. O conjunto de equações (2) podem ser escritas de modo mais compacto usando-se a notação matricial. Ter-se-á:

$$G \vec{x} = \vec{y}^c \tag{6}$$

Ond G é a matrix $\begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ 1 & z_3 \\ \dots & \dots \end{bmatrix}$ e \vec{x} o vector contendo os parêmtros a e b, $\vec{x} = \begin{bmatrix} a \\ b \end{bmatrix}$ e \vec{y}^c é o

vector que contém os valores da temperatura, $\vec{y}^c = \begin{bmatrix} T_1^c \\ T_2^c \\ T_3^c \\ \dots \end{bmatrix}$. A função objectivo será

$Q = \vec{e} \vec{e}^T$ sendo \vec{e} o vector (conhecido como vector erro) que contém as diferenças entre os valores observados e os calculados (T representa a operação de transposição), isto é,

$$\vec{e} = (\vec{y}^o - \vec{y}^c) = (\vec{y}^o - G \vec{x})$$

A minimização da função objectivo conduz a,

$$(G^T G) \vec{x} = G^T \vec{y}^o \tag{7}$$

A solução que permite o cálculo dos parâmetros é,

$$\vec{x} = (G^T G)^{-1} G^T \vec{y}^o \tag{8}$$

A equação (7) em notação matricial traduz de facto o sistema de equações expresso em (4) cuja solução é dada por (8).

Exemplo:

Considere os dados da tabela anexa, provenientes de uma experiência laboratorial.

- a) Represente graficamente o conjunto de valores;
- b) Elabore um script MatLab que lhe permita determinar os valores de A e B da recta $Y=A X + B$ que melhor representa o conjunto de dados, no sentido dos mínimos quadrados.
- c) Estime os erros associados às suas estimativas de A e B
- d) Represente num único gráfico a recta e os dados.

Xi (mV)	Fluxo Yi (l/s)
1.01	0.00
1.27	0.19
1.85	0.58
2.38	0.96
2.83	1.26
3.13	1.47
3.96	2.07
4.91	2.75

O script MatLab seguinte resolve o problema.

```

% ajuste de uma função a dados experimentais
% método dos mínimos desvios quadrados
%
% ajuste linear usando 3 metodos diferentes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; close all; clc;
% dados
X = [1.01;1.27;1.85;2.38;2.83;3.13;3.96;4.91];
Y = [0.00;0.19;0.58;0.96;1.26;1.47;2.07;2.75];
N = length(X);
figure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(3,1,1)
scatter(X,Y);
xlabel('X (mV)'); ylabel('Fluxo (l/s)');
xlim([0 5]); ylim([-1 3]);
hold on;
grid on;
a=(N*sum(X.*Y)-sum(X)*sum(Y))/(N*sum(X.^2)-(sum(X))^2);
b=(sum(Y)-a*sum(X))/N;
% recta obtida
Ymodl=a*X+b;
%erros
S_1 = sqrt(sum((Y-Ymodl).^2)/(N-2));
ErroA_1 = S_1*sqrt(N/((N*sum(X.^2)-(sum(X))^2)));
ErroB_1 = S_1*sqrt(sum(X.^2)/((N*sum(X.^2)-(sum(X))^2)));
plot(X,Ymodl);
title(['A = ' num2str(a) '+-' num2str(ErroA_1) ' ; B = '
num2str(b) '+-' num2str(ErroB_1)]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    
```

```

G = [ones(N,1) X];
Pest = (G'*G)^(-1)*G'*Y;

% recta obtida
Ymod2 = Pest(2)*X+Pest(1);

S_2 = sqrt(sum((Y-Ymod2).^2)/(N-2));

ErroA_2 = S_2*sqrt(N/((N*sum(X.^2))-(sum(X))^2));
ErroB_2 = S_2*sqrt(sum(X.^2)/((N*sum(X.^2))-(sum(X))^2));

subplot(3,1,2)
scatter(X,Y);
xlabel('X (mV)'); ylabel('Fluxo (1/s)');
xlim([0 5]); ylim([-1 3]);
hold on;
grid on;
plot(X,Ymod2,'g');
title(['A = ' num2str(Pest(2)) '+-' num2str(ErroA_2) ' ; B = '
' num2str(Pest(1)) '+-' num2str(ErroB_2)]);

%-----
%-----
% Mínimos quadrados com função do Matlab
p = polyfit(X,Y,1);

% recta obtida
Ymatlab = p(1)*X+p(2);

% ERRO
S_3 = sqrt(sum((Y-Ymatlab).^2)/(N-2));
ErroA_3 = S_3*sqrt(N/((N*sum(X.^2))-(sum(X))^2));
ErroB_3 = S_3*sqrt(sum(X.^2)/((N*sum(X.^2))-(sum(X))^2));

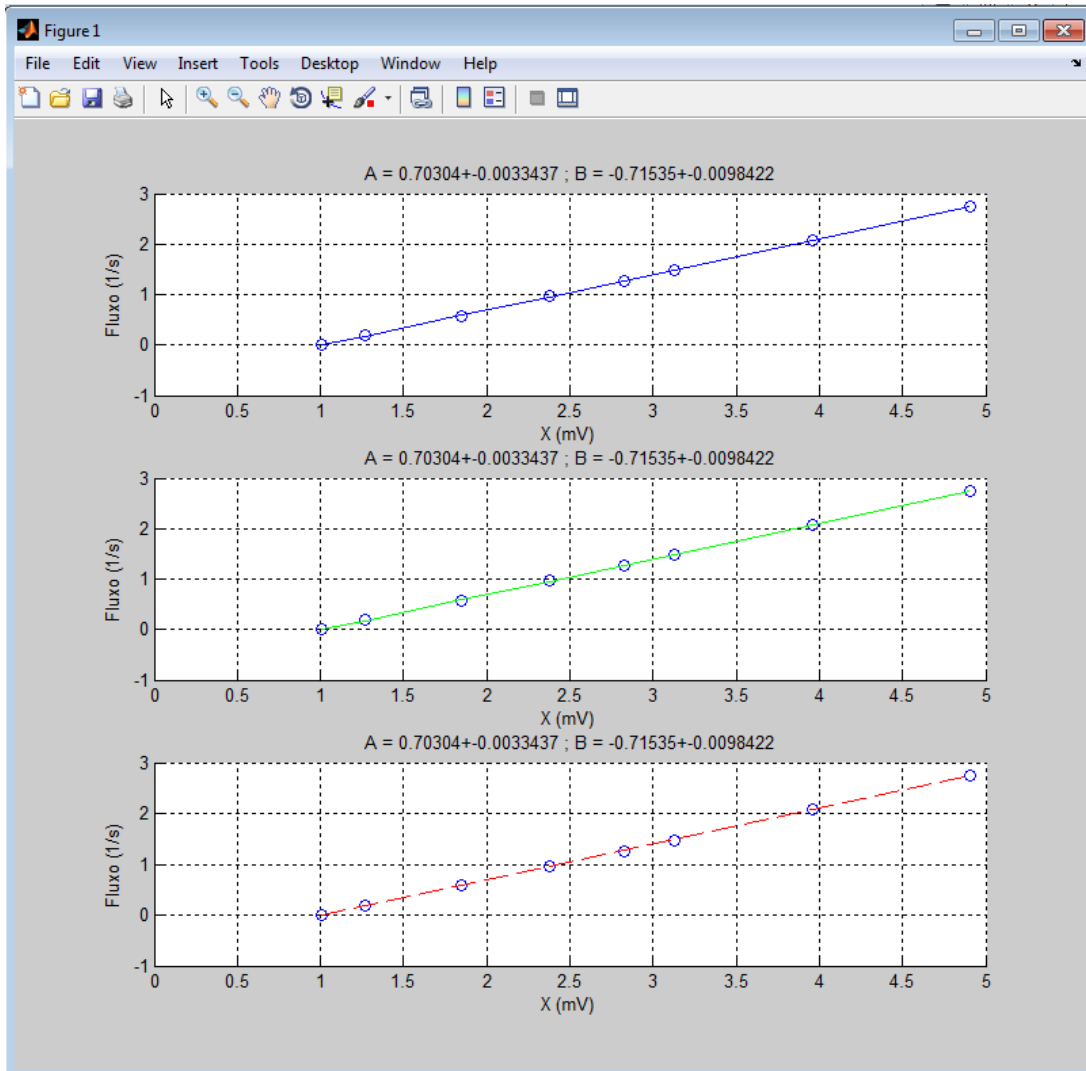
subplot(3,1,3)
scatter(X,Y);
xlabel('X (mV)'); ylabel('Fluxo (1/s)');
xlim([0 5]); ylim([-1 3]);
hold on;
grid on;

plot(X,Ymatlab,'r--');

title(['A = ' num2str(p(1)) '+-' num2str(ErroA_3) ' ; B = '
num2str(p(2)) '+-' num2str(ErroB_3)]);

```

Os gráficos resultantes são apresentados em seguida.



Erro da estimativa dos parâmetros

Os valores obtidos para a e b são estimativas dos verdadeiros valores. Importa pois, saber-se o erro com que aqueles valores foram determinados. Aplicando a regra da propagação dos erros tem-se, para a variância de y_i ,

$$\sigma^2(y_i) = \left(\frac{\partial y}{\partial a}\right)^2 \sigma^2(a) + \left(\frac{\partial y}{\partial b}\right)^2 \sigma^2(b)$$

$$\sigma^2(y_i) = \sigma^2 = \frac{1}{N-2} \sum (y_i - a - b z_i)^2$$

e

$$\sigma^2(a) = \sum \left(\frac{\partial a}{\partial y_i}\right)^2 \sigma^2(y_i)$$

$$\sigma^2(b) = \sum \left(\frac{\partial b}{\partial y_i} \right)^2 \sigma^2(y_i)$$

Calculando-se as derivadas, tendo em atenção as equações (5), obtêm-se para as variâncias,

$$\sigma^2(a) = \sigma^2 \frac{\sum z_i^2}{\Delta}$$

$$\sigma^2(b) = \sigma^2 \frac{N}{\Delta} \quad (9)$$

O erro na estimativa de cada um dos parâmetros será dados pela raíz quadrada destas grandezas.

Coefficiente de correlação

Na aplicação do método dos mínimos quadrados assume-se que há uma relação linear entre as variáveis y e x , sendo x conhecido exactamente, isto é,

$$y = a + b x \quad (10)$$

Se fôr o caso, então a relação linear é bijectiva e tem-se também,

$$x = a' + b' y \quad (11)$$

Tendo-se, do mesmo modo que anteriormente (equação 5),

$$b' = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{N \sum y_i^2 - (\sum y_i)^2}$$

Neste caso tem-se então,

$$y = -\frac{a'}{b'} + \frac{x}{b'}$$

Igualando esta equação com a equação (10), a igualdade só se verifica se

$$a = -\frac{a'}{b}$$

e

$$b b' = 1$$

Numa situação ideal a correlação será de 100 % ou muito próxima desse valor, isto é, o produto $(b b')$ será igual a 1. Quando se tratam dados reais há uma incerteza no cálculo dos coeficientes que afecta o valor daquele produto. Define-se então o coeficiente de correlação (R) do seguinte modo,

$$R = \sqrt{b \overline{b'}} \quad (12)$$

Deve notar-se, contudo, que este coeficiente não fornece nenhuma informação sobre a confiança com que o modelo linear pode ser usado na determinação dos parâmetros.

Exercício proposto:

Considere a tabela seguinte que contém valores da porosidade (P) e do factor de formação (F) de várias amostras de rochas. Assumindo que P e F estão relacionados por uma expressão da forma $F = a P^{-b}$,

- Estabeleça o sistema de equações que resulta da aplicação do método dos mínimos quadrados para a estimativa de a e b;
- Escreva um script MatLab que permita calcular os valores, os erros associados e o valor do coeficiente de correlação para este problema.

porosidade	F
0.45	3.92
0.43	4.1
0.37	4.77
0.38	4.88
0.242	8.72
0.257	8.2
0.285	6.1
0.329	5.6
0.23	9.6

TEMA 7

OS MÉTODOS CRS e SA

A interpretação dos dados adquiridos durante uma experiência constitui um dos mais importantes passos para o conhecimento dos fenómenos. Por *interpretação dos dados* entende-se aqui, a complexa (e difícil) tarefa de estabelecer um *modelo* que cumpra duas condições fundamentais: 1) que satisfaça o conjunto de dados recolhidos, de acordo com um critério pré-estabelecido, e 2) seja susceptível de ter interpretação física (ou outra, dependendo do estudo em questão) que esteja de acordo com informação conhecida. Neste contexto o termo *modelo* designa o conjunto de valores das propriedades relevantes (designados por *parâmetros do modelo*) que caracterizam o(s) fenómeno(s).

Formulação do problema

Seja então \mathbf{d}^o o vector contendo o conjunto dos valores observados (valores medidos ou calculados a partir de valores medidos). Cada elemento de \mathbf{d}^o pode ser descrito como o valor aproximado do *verdadeiro* valor da grandeza a medir \mathbf{d} , ao qual não se tem acesso, possuindo-se uma estimativa do erro da medida, tal que

$$\phi_d(\mathbf{d}^o, \mathbf{d}) \leq \varepsilon_d \quad \text{com a condição } \mathbf{d}^o \rightarrow \mathbf{d} \text{ quando } \varepsilon_d \rightarrow 0$$

onde ϕ representa uma distância (métrica) definida no espaço dos dados.

Seja \mathbf{m}^{est} o vector dos parâmetros do modelo adoptado para a interpretação dos dados. Neste problema o vector \mathbf{m}^{est} é o vector das incógnitas e a resolução do problema deverá permitir conhecer os elementos deste vector. De acordo com os métodos adoptados, para a resolução do problema, os valores de \mathbf{m}^{est} representam uma estimativa dos *verdadeiros* valores dos parâmetros \mathbf{m} , isto é,

$$\phi_m(\mathbf{m}^{\text{est}}, \mathbf{m}) \leq \varepsilon_m \quad \text{com a condição } \mathbf{m}^{\text{est}} \rightarrow \mathbf{m} \text{ quando } \varepsilon_m \rightarrow 0$$

ϕ_m representa uma distância definida no espaço dos parâmetros do modelo. Neste contexto o termo *modelo* é usado para designar as relações matemáticas que ligam os parâmetros \mathbf{m} à resposta do modelo \mathbf{d} ,

$$\mathbf{d} = g(\mathbf{m}).$$

De acordo com a natureza daquelas relações é usual classificar este tipo de problemas em lineares e não lineares.

Resolução do problema usando métodos de minimização global

Alguns dos métodos usados na resolução do problema exposto são adequados quando a topologia da função objectivo tem apenas um mínimo. Esta situação não é, contudo, a mais frequente, conduzindo a que a solução obtida dependa do modelo inicial e de um parâmetro de estabilização. Com o objectivo de evitar, ou minimizar, estas dependências recorre-se por vezes ao uso de métodos de resolução que podem ser considerados como métodos de *minimização global*. Analisam-se, em seguida, dois destes métodos: o “simulated annealing” e o método “controlled random search” (Price, 1977) que

traduzimos aqui por “procura aleatória controlada”. Estes métodos podem ser considerados como variantes do método de Monte Carlo. A ideia básica é a de investigar (aleatoriamente) o espaço de parâmetros (tão exaustivamente quanto possível) levando lentamente a investigação a centrar-se, mas não exclusivamente, em torno do mínimo absoluto. Para cada conjunto de parâmetros (modelo) determina-se o valor da função objectivo retendo-se os modelos que apresentem valores inferiores a um valor estabelecido. Uma análise estatística deste conjunto restrito de modelos permite a estimativa das funções de distribuição *a posteriori* dos parâmetros.

Os métodos de *minimização global* têm, relativamente aos métodos de *minimização local*, algumas vantagens. Entre estas assinala-se o facto de não ser necessário o conhecimento da matriz das sensibilidades, que é difícil de calcular para alguns modelos. A maior desvantagem reside no facto de consumirem muito tempo de cálculo.

“Simulated annealing (SA)”

O nome deste algoritmo traduz a semelhança existente entre o algoritmo de SA e o processo físico de “têmpera” de metais. O método inicia a procura da solução do problema explorando aleatoriamente o espaço de parâmetros. Lentamente o espaço de parâmetros explorado é restringido ao sub-espaço onde se encontra o mínimo absoluto. Este processo é semelhante ao arrefecimento do metal e o seu controlo é feito através de uma variável designada, nos algoritmos SA, por temperatura.

O algoritmo de Metropolis et al. (1953), o algoritmo inicial do SA, é um algoritmo iterativo. A iteração k inicia-se pela geração aleatória de uma solução (com base na solução da iteração $k-1$) \mathbf{m}^k e do cálculo da respectiva função objectiva ϕ^k . Se $\phi^k < \phi^{k-1}$ a solução é aceite e o processo iterativo continua pela geração de uma nova solução. Se $\phi^k > \phi^{k-1}$ a aceitação, ou não, da solução \mathbf{m}^k obriga ao cálculo da função $P(\Delta\phi) = \exp((\phi^{k-1} - \phi^k)/T^k)$ e à geração aleatória do número real r , tal que $r \in [0,1]$. A solução \mathbf{m}^k é aceite se $r < P(\Delta\phi)$ prosseguindo o processo iterativo. T é a temperatura, que no início do processo iterativo deve ser elevada, diminuindo (de acordo com uma lei estabelecida) ao longo do processo.

Nas fases iniciais, e porque T é elevado, são aceites soluções muito diferentes. À medida que T diminui a aceitação de soluções torna-se mais exigente e apenas as melhores soluções são aceites levando o algoritmo a convergir. O processo iterativo termina quando (por exemplo) a variação da função objectivo entre iterações é inferior a um valor preestabelecido.

Para o mesmo problema, o algoritmo SA pode ser corrido várias vezes, com condições iniciais diferentes, de modo a obter-se um conjunto (estatisticamente) significativo de soluções. A análise estatística deste conjunto de soluções permite calcular o modelo médio $\langle \mathbf{m} \rangle$ e a matriz de covariância *a posteriori* dos parâmetros do modelo usando, respectivamente (Mrinal et al., 1993):

$$\langle \mathbf{m} \rangle = \sum \mathbf{m} P(\mathbf{m}) \quad (7.1.1)$$

$$\mathbf{C} = \sum (\mathbf{m} - \langle \mathbf{m} \rangle) (\mathbf{m} - \langle \mathbf{m} \rangle)^T P(\mathbf{m}) \quad (7.1.2)$$

onde as somas são sobre os modelos estimados e $P(\mathbf{m})$ é a densidade de probabilidade *a posteriori*,

$$P(\mathbf{m}) \propto \exp(\phi(\mathbf{m})) \quad (7.1.3)$$

Exercício:

O ficheiro 'K.mat' contém uma série empírica $K(q)$ (e o respetivo q) que pode ser modelada por uma expressão da forma:

Recorrendo a um método SA estime os parâmetros α e C_1 que se ajustam a estes dados, sabendo que os parâmetros estarão numa gama de valores entre 0 e 2 sendo $\alpha \neq 1$

```

% Simulated annealing - ex.1

% Este é um metodo de optimizacao para o calculo dos parametros que melhor
% ajustam uma lei a um conjunto de dados empiricos.
% Começamos com um modelo aleatorio inicial (um alfa e um C1 aleatorios) e
% vamos perturbando (ou seja, fazendo incrementos) a cada um deles. Para
% cada novo modelo perturbado que e calculado calculamos tambem a energia e
% verificamos se esta e inferior a energia do modelo anterior. Se
% for, substituímos o modelo anterior pelo perturbado. No entanto, uma condicao
% a mais e imposta para permitir que por vezes sejam aceites modelos piores
% que o anterior. Isto permite que a busca possa estender-se a todas as
% regioes do espaço de parametros, e nao se concentre apenas perto de uma
% zona que possa corresponder apenas a um minimo local, e nao global, do
% erro. Para valores altos do parametro T (analogia com temperatura no
% annealing da metalurgia) isto acontece muitas vezes (exp(-dE/T)>rand
% quase sempre verificada), o que permite que a busca inicial seja sobre
% todo o espaço. No entanto, com a diminuicao de T, aquela condicao
% torna-se mais dificil de verificar e a busca concentra-se na zona do
% minimo mais global para onde entretanto se convergiu.

clc; clear all; close all;

aux = load('Kemp.mat'); %abre o ficheiro de dados do matlab
Kemp = aux.K; %variável Kemp guarda valores conhecidos como K
dentro do ficheiro
q = aux.q; %variável q guarda valores conhecidos como q dentro
do ficheiro

salto = (2-0)/3; %valor maximo da perturbacao
T = 1e2; %"temperatura" inicial

Erro_Max = 1; %erro maximo a ser admitido
Iter_Max = 2000; %num max de iteracoes (este e o anterior vao
determinar a temperatura minima a ser atingida
niter = 100; %num de modelos perturbados a serem calculados a
cada temperatura

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solução Inicial aleatória

alfa = 2*rand(1); %gera um num aleatorio entre 0 e 2
C1 = 2*rand(1); %gera um num aleatorio entre 0 e 2
if (alfa == 1) %garante que alfa!=1
    alfa = 2*rand(1);
end

K = (C1/(alfa-1))*(q.^alfa-q); %calcula o modelo que corresponde aqueles
parametros
E = sum(sqrt((K-Kemp).^2)); %calcula a energia (erro) correspondente a
esse modelo ???ERRO

figure %plot da solução inicial aleatória, em
conjunto com os dados empiricos. Nota que cada vez que corres o programa e
gerado um modelo aleatorio diferente
scatter(q,Kemp);
hold on;
plot(q,K);
title(['alfa=' num2str(alfa) '; C1=' num2str(C1)]);
ylim([-1 3]);
xlabel('q'); ylabel('K(q)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%Ciclo que vai levar a convergencia:
iter = 0;
while (E>Erro_Max) && (iter < Iter_Max)           %duas condicoes que vao
determinar até quando se deve baixar a temperatura: para a busca quando energia
superior a Emax ou quando num max iteracoes for atingido

    iter = iter+1;                               %contador do num de
iteracoes

    for j = 1:niter                               %calculo de niter modelos
perturbados

        pert = salto*rand(1)*(-1)^(round(rand(1))); %valor da perturbacao que
vai ser feita a alfa: +ou-(salto*rand): valor max da perturbacao é salto. (-
1)^round(rand) da 1 quando rand entre 0 e 0.5 e -1 quando rand entre 0.5 e 1. round
retorna inteiro mais proximo
        alfa_pert = alfa+pert;                   %calculo de alfa perturbado
        if (alfa_pert == 1)                       %garantir que alfa=1
            pert = salto*rand(1)*(-1)^(round(rand(1)));
            alfa_pert = alfa+pert;
        elseif (alfa_pert < 0)                   %condicoes para nao
permitir que alfa_pert saia do dominio 0 a 2.
            alfa_pert = alfa+2*abs(pert);
        elseif (alfa_pert > 2)
            alfa_pert = alfa-2*abs(pert);
        end

        pert = salto*rand(1)*(-1)^(round(rand(1))); %perturbacao de C1: o mesmo
que para alfa, mas com uma perturbacao aleatoria diferente
        C1_pert = C1+pert;
        if (C1_pert < 0)
            C1_pert = C1+2*abs(pert);
        elseif (C1_pert > 2)
            C1_pert = C1-2*abs(pert);
        end

        K_pert = (C1_pert/(alfa_pert-1))*(q.^alfa_pert-q); %calculo do modelo
perturbado (com os novos parametros perturbados)
        Eold = E;                                 %a energia do ultimo modelo
a ser aceite passa a ser Eold
        E_pert = sum(sqrt((K_pert-Kemp).^2));     %calculo da energia
correspondente ao novo modelo perturbado
        dE = E_pert-Eold;                         %calculo da diferenca entre
as energias do modelo velho (ultimo aceite) e do modelo novo. Queremos aceitar modelos
com energia inferior ao anterior

        if (dE <= 0) || (exp(-dE/T) > rand(1)) %primeira condicao e
simples: passa se modelo novo for melhor que antigo. a segunda permite que alguns
(inicialmente, para Ts elevados, muitos) modelos piores sejam aceites e a busca possa
ser mais generalizada. À medida que T baixa, a segunda condicao e cada vez mais dificil
de verificar e busca concentra-se na vizinhanca de um minimo, que esperamos que seja o
global
            alfa = alfa_pert;                     %caso uma das condições
seja verificada, substituímos entao o modelo antigo, guardado, pelo novo modelo
perturbado
            C1 = C1_pert;                         %guardamos os novos
parametros, o novo modelo e a energia correspondente
            K = K_pert;
            E = E_pert;
        end

    end

    T = 0.9*T;                                   %no fim de se calcularem os
niter modelos perturbados baixamos a "temperatura", o que vai restringir a quantidade
de modelos potencialmente piores que passam. Para Ts baixos, vai localizar mais a busca
em torno de um so minimo (que em principio sera o global)

end                                               %no fim do while teremos o
modelo otimizado, correspondente a um minimo global da funcao de erro (funcao
objetivo dE minimizada)

figure                                           %representação dos dados
empiricos e do modelo otimizado
scatter(q,Kemp)
hold on;
plot(q,K);
ylim([-1 3]);
xlabel('q'); ylabel('K(q)');
title(['alfa=' num2str(alfa) ' ; C1=' num2str(C1)]);
disp(['última temperatura = ' num2str(T) ' K; em ' num2str(iter) ' ciclos de T']);

```

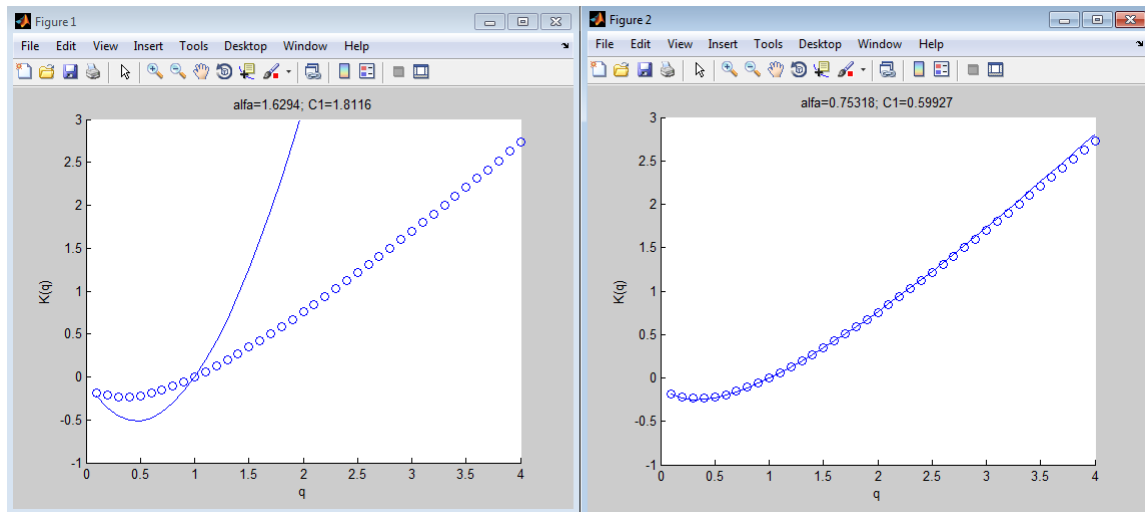


Figura: solução inicial e final para o exercício1, usando-se o métodos SA.

“Controlled random search (CRS)” -Procura aleatória controlada

Na aplicação do algoritmo (iterativo) de Price (1977) principia-se pela geração aleatória de L modelos com a condição de que os parâmetros pertençam a um subdomínio do domínio (permitido) dos parâmetros. Isto significa que os parâmetros têm um limite inferior e superior,

$$\mathbf{m}_l \leq \mathbf{m}^{est} \leq \mathbf{m}_s . \tag{7.2.1}$$

Percebe-se facilmente que quanto maior for L mais exaustiva se torna a procura do mínimo absoluto. Calcula-se a função objectivo ϕ para cada modelo identificando-se o modelo \mathbf{m}_{max} que apresentar o valor máximo para $\phi = \phi_{max}$. Na iteração k, um novo modelo \mathbf{m}^k é estimado tendo-se em atenção os parâmetros de (M+1) modelos escolhidos aleatoriamente entre os L modelos. O vector dos parâmetros do novo modelo é dado por,

$$\mathbf{m}^k = 2 \mathbf{c} - \mathbf{m}_{M+1} \tag{7.2.2}$$

onde \mathbf{c} é o centróide dos M primeiros modelos escolhidos, \mathbf{m}_{M+1} é o vector dos parâmetros correspondente ao (M+1)-ésimo modelo escolhido sendo M o número de parâmetros a determinar. Se os elementos de \mathbf{m}^k obedecerem aos constrangimentos impostos e se o valor da correspondente função objectivo for inferior a ϕ_{max} o anterior modelo \mathbf{m}_{max} é retirado dos L modelos iniciais e substituído pelo novo modelo. Se o novo modelo não satisfizer os constrangimentos ou se a correspondente função objectivo for maior que ϕ_{max} , o novo modelo não é considerado procedendo-se a nova selecção aleatória de (M+1) modelos. A substituição de um modelo de L por um novo modelo constitui uma iteração. O processo iterativo termina quando as funções objectivo de todos os L modelos tiverem um valor inferior a um valor definido ϵ .

O conjunto de L possíveis soluções estimadas pelo método CRS formam um “cluster” de modelos na vizinhança do mínimo absoluto. O modelo representado pelo centróide deste “cluster”, \mathbf{m}^* , deverá ser mais significativo que qualquer dos modelos do conjunto e é considerado como a melhor estimativa dos parâmetros do modelo. A matriz de covariância dos parâmetros pode ser estimada por (Silva and Hohmann, 1983),

$$C_{ij} = \frac{1}{L-1} \sum_{k=1}^L (m_{ik} - m_i^*)(m_{jk} - m_j^*) \quad (7.2.3)$$

onde m_{ik} representa a k -ésima estimativa do parâmetro i e m_i^* é a melhor estimativa do parâmetro i , dada por

$$m_i^* = \frac{1}{L} \sum_{k=1}^L m_{ik}^*. \quad (7.2.4)$$

A matriz de correlação \mathbf{R} é obtida normalizando-se \mathbf{C} ,

$$R_{ij} = C_{ij} / (C_{ii} C_{jj})^{1/2} \quad (7.2.5)$$

```

% Controlled Random Search
clc; clear all; close all;

% aux = load('Kemp.mat','-ascii');
% Kemp = aux(1,:);
% q = aux(2,:);

aux = load('Kemp.mat');
Kemp = aux.K;
q = aux.q;

L = 20;
Eps = 1.0;
it_max = 5000;

%%%%%%%% L soluções Iniciais %%%%%%%%%

alfa = 2*rand(1,L);
C1 = 2*rand(1,L);
if (alfa == 1)
    alfa = 2*rand(1,L);
end

for i = 1:L
    % i varia o modelo
    for j = 1:length(q)
        % j é o índice da variável independente q
        K(j,i) = (C1(i)/(alfa(i)-1))*(q(j).^alfa(i)-q(j)); % calcula um
        modelo para cada conjunto de parâmetros
    end

    E(i) = sum(sqrt((K(:,i)-Kemp(:,1)).^2)); % calcula
    energia para cada modelo
end

%%%%%%%%
%%%%%%%%

```



```

it=0;

E_max = 1e10;
while ( (it < it_max) && (E_max > Eps) )

    it = it+1;                                %num iterações

    for i = 1:L;                                %calcula qual o modelo com energia máxima e o índice
        correspondente
        if E(i) == max(E)
            i_max = i;
            E_max = max(E);
        end
        if E(i) == min(E)                    %calcula energia mínima (para plot inicial)
            i_min = i;
        end
    end

    if it == 1                                %plot do melhor e pior modelos iniciais

        figure
        subplot 211
        scatter(q,Kemp);
        hold on;
        plot(q(:),K(:,i_max),'r');
        hold on;
        plot(q(:),K(:,i_min),'b');

        ylim([-1 3]);
        xlabel('q'); ylabel('K(q)');

    end

    i_3 = round(L*rand(1,3));                    %3 índices aleatórios de 1 a 20 e
    diferentes entre si
    while ( i_3(1)==0 || i_3(2)==0 || i_3(3)==0 || i_3(1)==i_3(2) ||
            i_3(1)==i_3(3) || i_3(2)==i_3(3) )
        i_3 = round(L*rand(1,3));
    end

    alfa_mean = mean(alfa(i_3(:)));                %alfa médio dos 3 alfas aleatórios
    C1_mean = mean(C1(i_3(:)));                    %C1 médio dos 3 C1s aleatórios

    K_mean = (C1_mean/(alfa_mean-1))*(q.^alfa_mean-q);

    E_mean = sum(sqrt((K_mean-Kemp).^2));

    if E_mean < E_max                            %compara a energia do novo modelo com a
        max anterior
        alfa(i_max) = alfa_mean;                    %substitui o modelo antigo pelo novo
        C1(i_max) = C1_mean;

        K(:,i_max) = K_mean;
        E(i_max) = E_mean;
    end
    it
end

%plot da solução otimizada: o modelo de energia mínima de entre o conjunto
final
subplot 212
scatter(q,Kemp);
hold on;
plot(q(:),K(:,i_min),'b');

ylim([-1 3]);
xlabel('q'); ylabel('K(q)');
title(['Solução otimizada com CRS: alfa=' num2str(alfa(i_min)) ', C1='
num2str(C1(i_min)) '. N° iterações: ' num2str(it) ] )

```

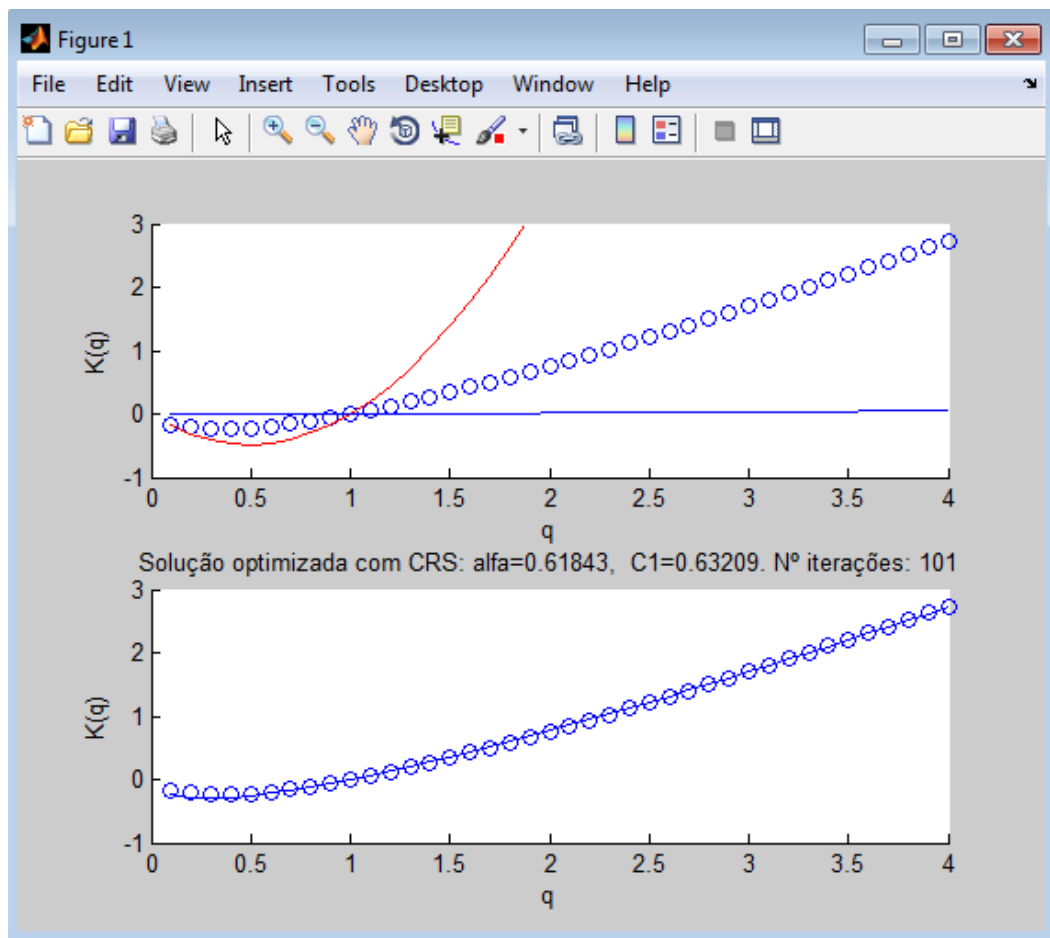


Figura: solução inicial e final para o exercício1, usando-se o métodos CRS.