

Ejercicio

Cádiz, 12 de Junio de 2008

Enunciado

Dada una serie de datos mediante dos archivos *x.dat* y *y.dat* encontrar los coeficientes *a* y *b* que minimizan el error medio cuadrático de la función

$$y = \frac{1}{a + be^{-x}}$$

Pasos intermedios

1. Cargar los datos mediante el comando *load*, *x.dat* en la variable *x* e *y.dat* en la variable *y*
2. Definir la función *leasqrfunc*

```
function y = leasqrfunc(x,p)
y=1./(p(1)+p(2).*exp(-x));
```

3. Definir la función *leasqrdp*

```
function y = leasqrdp(x,f,p,dp,func)
y = [-1./(p(2).*exp(-x)+p(1)).^2, -exp(-x)./(p(2).*exp(-x)+p(1)).^2];
```

4. Resolver el problema de mínimos cuadrados mediante el algoritmo *Levenberg-Marquardt* implementado en la función *leasqr* del paquete de octave-forge *Optim*

Código

```
function fit
% generate test data
t = load('x.dat');
data = load('y.dat');
sorted = sortrows([t,data],1);

t=sorted(:,1);
data=sorted(:,2);

wt1 = (1 + 0 * t) ./ sqrt (data);

% Note by Thomas Walter <walter@pctc.chemie.uni-erlangen.de>:
%
```

```

% Using a step size of 1 to calculate the derivative is WRONG !!!!
% See numerical mathbooks why.
% A derivative calculated from central differences need: s
%     step = 0.001...1.0e-8
% And onesided derivative needs:
%     step = 1.0e-5...1.0e-8 and may be still wrong

F = @leasqrfunc;
dFdp = @leasqrdfdp; % exact derivative
% dFdp = @dfd;      % estimated derivative
dp = [0.001; 0.001];
pin = [0.5; 0.5];
stol=0.001; niter=100;
minstep = [0.01; 0.01];
maxstep = [0.8; 0.8];
options = [minstep, maxstep];

global verbose;
verbose=1;
[f1, p1, kvg1, iter1, corp1, covp1, covr1, stdresid1, Z1, r21] = ...
    leasqr (t, data, pin, F, stol, niter, wt1, dp, dFdp, options);

```