

MASTER MISV 2009–2010

## OPTIMISATION ET ALGORITHMIQUE

### Projet SCILAB surveillé, durée 2h

#### Problème :

On s'intéresse à l'ajustement de données au modèle paramétrique suivant  $\varphi(t) = x_3 e^{x_1 t} + x_4 e^{x_2 t}$  où la variable  $t \in \mathbb{R}_+$  et les paramètres du modèle  $x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4$ .

Les données  $(t_i, y_i)_{1 \leq i \leq m}$  sont disponibles dans le tableau `data`.

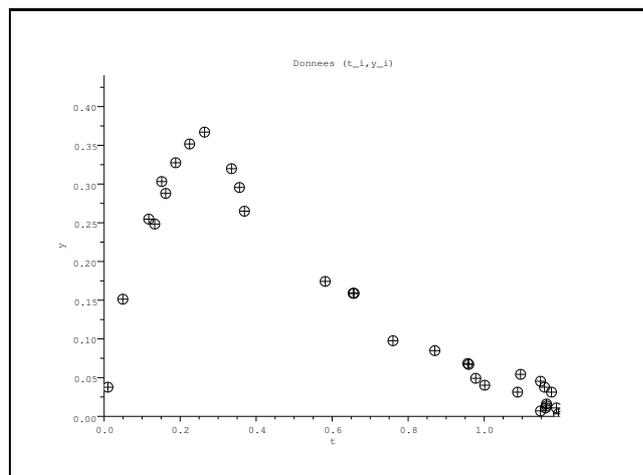
On se propose d'adapter la méthode de *région de confiance* avec l'algorithme *dogleg* pour déterminer  $x^* = \underset{x \in \mathbb{R}^4}{\operatorname{Argmin}} f(x)$  par moindres carrés non linéaires :

$$f(x) = \frac{1}{2} \sum_{k=1}^m r_k(x)^2 = \frac{1}{2} \|r(x)\|_2^2, \text{ où } r(x) = (r_1(x) \cdots r_m(x))^t \in \mathbb{R}^m, x \in \mathbb{R}^4.$$

On note  $J(x) \in \mathcal{M}(m, 4)$  la matrice jacobienne de  $r$  au point  $x$ .

Avant de commencer, copiez le fichier `tp_projet_09.sci` dans votre répertoire de travail.

1. Pour  $k = 1, \dots, m$  et  $x \in \mathbb{R}^4$ , donner la définition de  $r_k(x)$ .  
Écrire  $f(x)$  et déterminer  $J(x)$ .
2. En Scilab, écrire fonction `y = phi(x,t)` qui calcule  $\varphi(t)$  pour  $x$  donné.  
Et fonction `[r , J] = r_J(x,data)`, qui détermine  $r(x)$  et  $J(x)$  pour  $(t_i, y_i)_{1 \leq i \leq m}$  données.
3. Dans fonction `[xd] = dogleg(Delta,B,g)`, on est amené à résoudre le système linéaire  $B * x_B = -g$ . Afin de prévenir des problèmes de stabilité numérique, on va remplacer la ligne 9 par une résolution au sens des moindres carrés linéaires.  
Proposez une solution à ce problème et reprogrammez la fonction.
4. En inspectant les données, justifiez pourquoi on peut supposer  $x_1 < 0$ ,  $x_2 < 0$  et  $x_4 = -x_3$ .



Dans le fichier on propose `x0 = [ -1; -2; 1; -1]`. Déterminez alors  $x^*$ .

Essayez d'autres valeurs pour `x0`, `Delta` et `eta`. Que constatez vous ?

#### À la fin de la séance :

- (1) remettre un rapport avec : les réponses, commentaires
- (2) envoyer les fichiers sources, commandes et les résultats éventuels à [georges.koepfler@mi.parisdescartes.fr](mailto:georges.koepfler@mi.parisdescartes.fr)

```

1  function [xd] = dogleg(Delta,B,g)
2  delta = g'*B*g
3  xg = - ((g'*g)/delta) * g
4  if norm(xg) >= Delta then
5      xd = (Delta/norm(xg)) * xg
6      return
7  end
8  // Remplacer
9  xB = - inv(B)*g
10 // par
11 // xB =
12 if norm(xB) <= Delta then
13     xd = xB
14     return
15 end
16 a = (xB-xg)'*(xB-xg)
17 bp = xg'*(xB-xg)
18 c = xg'*xg-Delta^2
19 bet = (-bp + sqrt(bp^2-a*c))/a
20 xd = xg + bet*(xB-xg)
21 endfunction
22
23 function [x,k] = region_conf(x0,Delta_max,Delta_0,eta)
24 MAX_ITER=100;
25 precision=1e-10;
26
27 :
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74 function [r , J] = r_J(x,data)
75 t = data(:,1)
76 y = data(:,2)
77 // calcul de r et J
78 // ...
79 // r =
80 // J =
81 endfunction
82
83 function y = phi(x,t)
84 // y =
85 endfunction
86
87 function demo()
88 data = donnees()
89
90 scf();clf();
91 plot2d(data(:,1),data(:,2),style=-3,rect=[0,0,1.2,max(data(:,2))*1.2])
92 xtitle("Donnees (t_i,y_i)","t","y")
93
94 x0 = [ -1; -2; 1; -1]
95 Delta_max = 3 ; Delta_0 = 1 ;eta = 0.2
96 [x,k] = region_conf(x0,Delta_max,Delta_0,eta)
97
98 xx = linspace(0,1.2,100)';
99 plot2d(xx,phi(x,xx),style=[5])
100 titre = "Regression non lineaire : x1="+string(x(1))+", x2="+string(x(2))..
101         "+", x3="+string(x(3))+", x4= "+string(x(4))
102 xtitle(titre)
103 endfunction

```