

# Résolution d'un problème d'optimisation différentiable

Marc BOURQUI

Victor CONSTANTIN  
Florian SIMOND

Ian SCHORI

January 3, 2013

## Énoncé du problème

Trouver (une approximation de) la solution du problème suivant en appliquant le théorème de la plus forte pente:

$$\min_{x \in \mathbb{R}^2} (x_1 - 2)^4 + (x_1 - 2)^2 x_2^2 + (x_2 + 1)^2 \quad (1)$$

## Réponses aux questions

- (a) Implémenter la méthode de plus forte pente (Algorithme 11.3) à l'aide du logiciel MATLAB. Déterminer la taille du pas en appliquant la recherche linéaire, Algorithme 11.2 (les deux conditions de Wolfe).

Listing 1: pfp.m

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %
3 % Methodes de descente pour l'optimisation non lineaire %
4 % sans contraintes %
5 % %
6 % BOURQUI Marc %
7 % CONSTANTIN Victor %
8 % SCHORI Ian %
9 % SIMOND Floriant %
10 % %
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 function x = pfp(f, x0, alpha, useRL)
13
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 % Interface %
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18 % nom de la fonction a minimiser, qui est specifiee dans ...
19 % le fichier 'f.m'
20 fct = f;
21
22 % point initial
23 x = x0;
24
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 % Parametres %
27 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
28
29 % pour le critere d'arret
30 epsilon = 0.001 ;
31 maxIter = 200 ;
32
33 % initialisation du nombre d'iterations
```

```

34 i=1 ;
35
36 % initialisation de la matrice qui stocke tous les iterés
37 % un iteré = une colonne de cette matrice
38
39 stock(:,i) = x0;
40
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42 % Boucle principale %
43 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
44
45
46 % Critere d'arret: x a atteint la precision demandée OU nb ...
    iterations max atteint
47
48 while ( normGradient(fct,stock(:,i)) >= epsilon ) && ( i < ...
    maxIter )
49     % mise a jour du nombre d'iterations
50     i = i+1;
51
52     fprintf('Iteration number %d : x = [%f, %f]\n', i, ...
        stock(:,i-1)(1), stock(:,i-1)(2));
53     % calcul et stockage de la valeur du nouveau x
54     stock(:,i) = pfpInnerLoop(fct, stock(:,i-1), alpha, ...
        useRL);
55
56 end
57
58 % Calcul de la taille de la matrice contenant tous les x
59 taille = size(stock,2);
60
61 % Evaluation de la fonction en chaque point
62 for i=1:taille
63     valeurstock(i)=feval(fct,stock(:,i));
64 end
65
66
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 % Affichage des résultats %
69 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70
71 disp('Valeur de la suite des x :') ;
72 stock '
73
74 disp('*****')
75 disp(['Nombre d''iterations : ' ...
    num2str(i-1)])
76 disp(['Valeur de la fonction a l''optimum : ' ...
    num2str(feval(fct,stock(:,i)))]) ;
77 disp('Valeur de l''optimum : ')
78 xOptim = stock(:,i) '
79 disp('*****')
80

```

```

81 % passage au module de visualisation de la fonction et des ...
    resultats
82
83 visual3d(fct , stock , valeurstock);
84
85 sprintf('Nombre de fois que la boucle a ete parcourue : ...
    %d',i)
86
87 clear;
88 end

```

Listing 2: pfpInnerLoop.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2
3  %                                                                    %
4
5  % Calcul d'un itéré et du pas soit en utilisant                    %
6
7  % la recherche linéaire soit la formule de Cauchy                %
8
9  %                                                                    %
10
11 % BOURQUI Marc                                                    %
12
13 % CONSTANTIN Victor                                              %
14
15 % SCHORI Ian                                                      %
16
17 % SIMOND Floriant                                                %
18
19 %                                                                    %
20
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22
23
24
25 function x = pfpInnerLoop(f , x0 , alpha , useRL)
26
27     x = x0;
28
29
30
31     [fx , gfx] = feval(f , x);
32
33     d = -gfx;
34
35
36
37     % Calcul du pas
38

```

```

39     if useRL
40
41         beta1 = 0.5;
42
43         beta2 = 0.75;
44
45         lambda = 2;
46
47         alpha = rl(f, x, fx, gfx, alpha, beta1, beta2, ...
48                     lambda);
49
50     else
51
52         %Soit on peut utiliser la fonction dans b) pour ...
53         %calculer le pas
54
55         alpha = tp(f,x);
56
57     end
58 end

```

Listing 3: rl.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Implémente la recherche linéaire %
3  %                                %
4  % BOURQUI Marc                   %
5  % CONSTANTIN Victor              %
6  % SCHORI Ian                     %
7  % SIMOND Floriant                %
8  %                                %
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10
11 function alpha = rl(f, x, d, alpha0, beta1, beta2, lambda)
12     alpha = alpha0;
13     alphas = 0;
14     alphas = inf;
15
16     [fxad, fgxad] = feval(f, x + alpha * -gfx);
17
18     while (fxad > fx + alpha * beta1 * gfx' * -gfx) || ...
19           (fgxad' * -gfx < beta2 * gfx' * -gfx)
20         if fxad > fx + alpha * beta1 * gfx' * -gfx
21             alphas = alpha;
22             alpha = (alphas + alphas)/2;
23         elseif fgxad' * -gfx < beta2 * gfx' * -gfx
24             alphas = alpha;
25             if alphas < inf
26                 alpha = (alphas + alphas)/2;
27             else

```

```

27         alpha = lambda * alpha ;
28     end
29 end
30
31     [fxad , fgxad] = feval(f, x + alpha * -gfx);
32 end
33 end

```

- (b) Implémenter une fonction qui donne la taille du pas suivant:

$$\alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)} \quad (2)$$

Quelle est la nature de ce pas? D'où cette formule vient-elle?

- (c) Comparer le comportement de l'algorithme en utilisant les pas (a) et (b).
- (d) Comparer la methode de plus forte pente et la methode quasi-Newton (qui est déjà implementée – Série 3).