

# Optimisation et contrôle - TP

Constantini et Martin

Avril 2016

## 1 Problème primal

### 1.1 Questions théoriques

#### 1.1.1 Équivalence des problèmes

Montrons que les problèmes (13), (14) et (19) sont équivalents.

(13)  $\implies$  (14) : Soit  $q^s \in \mathbb{R}^n$  et  $f_r^s \in \mathbb{R}^{m_r}$  solutions de (13).

Soit  $q \in \mathbb{R}^n$ . Posons  $f_r = A_r q$ . On a alors

$$\begin{aligned} \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle &= \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, f_r \rangle \\ &\geq \frac{1}{3} \langle q^s, r \bullet q^s \bullet |q^s| \rangle + \langle p_r, f_r^s \rangle \\ &= \frac{1}{3} \langle q^s, r \bullet q^s \bullet |q^s| \rangle + \langle p_r, A_r q^s \rangle \end{aligned} \quad (1)$$

en utilisant les propriétés (13.a) et (13.b). Ainsi,  $q^s$  et  $f_r^s$  sont solutions du problème (14.a) et (13.b)  $\implies$  (14.b).

Donc (13)  $\implies$  (14)

(14)  $\implies$  (19) :

Soit  $q^s = \begin{bmatrix} q_T^s \\ q_C^s \end{bmatrix} \in \mathbb{R}^n$ . On suppose que  $q^s$  est solution de (14). On a donc, d'après (14.b),  $q^s = q^{(0)} + Bq_C^s$

$$\frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle \geq \frac{1}{3} \langle q^s, r \bullet q^s \bullet |q^s| \rangle \quad (2)$$

Ce qui correspond au problème (19). Donc (14)  $\implies$  (19).

(19)  $\implies$  (13): Soit  $q_C^s \in \mathbb{R}^{n-m_d}$  solution du problème (19). Posons  $q^s = q^{(0)} + Bq_C^s$  et  $f^s = \begin{bmatrix} f_r^s \\ f_d \end{bmatrix}$  avec  $f_r^s = A_r q^s$ . En utilisant la relation (18), on obtient,

$$\begin{aligned} A_d q^s - f_d &= A_d q^{(0)} + A_d B q_C^s - f_d \\ &= A_{d,T} A_{d,T}^{-1} f_d + A_{d,C} 0_{n-m_d} - A_{d,T} A_{d,T}^{-1} A_{d,C} q_C^s + A_{d,C} I_{n-m_d} q_C^s - f_d \\ &= 0 \end{aligned} \quad (3)$$

Ainsi,  $q^s$  et  $f^s$  sont solutions du problèmes (13.b)

Il reste à montrer que  $q^s$  et  $f^s$  satisfont (13.a).

Nous effectuons le raisonnement inverse qu'au début. Soit  $q \in \mathbb{R}^n$  et  $f = \begin{bmatrix} f_r \\ f_d \end{bmatrix}$  tel que  $Aq - f = 0$ .

$$\begin{aligned} \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, f_r \rangle &= \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle \\ &\geq \frac{1}{3} \langle q^s, r \bullet q^s \bullet |q^s| \rangle + \langle p_r, A_r q^s \rangle \\ &= \frac{1}{3} \langle q^s, r \bullet q^s \bullet |q^s| \rangle + \langle p_r, f_r^s \rangle \end{aligned} \quad (4)$$

Ainsi, (19)  $\implies$  (13).

Nous obtenons donc l'équivalence des trois problèmes.

### 1.1.2 Existence et unicité de la solution

Le problème (14) s'écrit de la manière suivante:

$$\min_{q \in U^{ad}} J(q) \quad (5)$$

avec  $U_d d = q$ ;  $A_d q - f_d = 0$  et  $J(q) = \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle$

J est de classe  $C^2$ . D'après l'expression de la hessienne écrite dans la partie ci-dessous, H est semi-définie positive et J est donc convexe. Cependant, il faut démontrer la stricte convexité et non la convexité simple.

Montrons d'abord l'existence d'une solution au problème de minimisation.  $U^{ad}$  est fermé car  $A_d^{-1}(f_d)$  fermé ( $A_d$  continue et singleton fermé).  $U^{ad}$  est non vide car  $A_d$  est surjective.  $U^{ad}$  est convexe comme image réciproque du convexe  $f_d$  par une application linéaire.

Puis J est coercive (Equivalence des normes en dimensions finies et Cauchy-Schwarz). Donc, J convexe, s.c.i et coercive sur  $U^{ad}$  fermé convexe et ainsi, par le théorème 5.3, le problème (14) admet au moins une solution.

Pour montrer qu'elle est unique, il faut montrer la stricte convexité que nous ne montrons pas ici.

On montre l'existence et unicité du problème (19) de la même manière.

## 1.2 Calcul du gradient et de la hessienne

Le problème de minimisation est le suivant:

$$\min_{q_c \in \mathbb{R}^{n-m_d}} \frac{1}{3} \langle q^{(0)} + Bq_c, r \bullet (q^{(0)} + Bq_c) \bullet |q^{(0)} + Bq_c| \rangle + \langle p_r, A_r(q^{(0)} + Bq_c) \rangle \quad (6)$$

D'où,

$$F(q_c) = \frac{1}{3} \langle q^{(0)} + Bq_c, r \bullet (q^{(0)} + Bq_c) \bullet |q^{(0)} + Bq_c| \rangle + \langle p_r, A_r(q^{(0)} + Bq_c) \rangle \quad (7)$$

Calculons maintenant le gradient:

$$G(q_c) = B^T(r \bullet (q^{(0)} + Bq_c) \bullet |q^{(0)} + Bq_c|) + (A_r B)^T p_r \quad (8)$$

Puis le hessien:

$$H(q_c) = 2B^T \text{diag}(r \bullet |q^{(0)} + Bq_c|)B \quad (9)$$

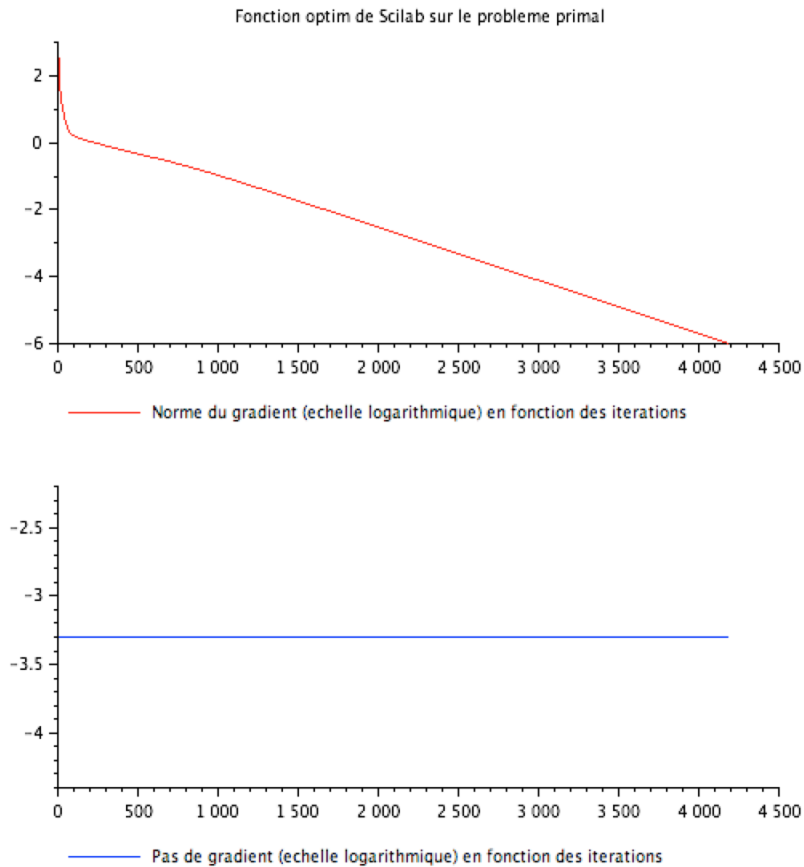
## 1.3 Différentes méthodes

### 1.3.1 Gradient à pas fixe

Après avoir rédigé la fonction **OraclePG** sur Scilab, nous avons testé cet oracle. En entrée de la fonction **Gradient\_F**, nous trouvons donc  $x_{ini}$ , un vecteur aléatoire de dimension  $n - md$ , et l'oracle.

Iteration	4182
Temps CPU	0.236469
Critere optimal	-3.734007
Norme du gradient	0.0000010

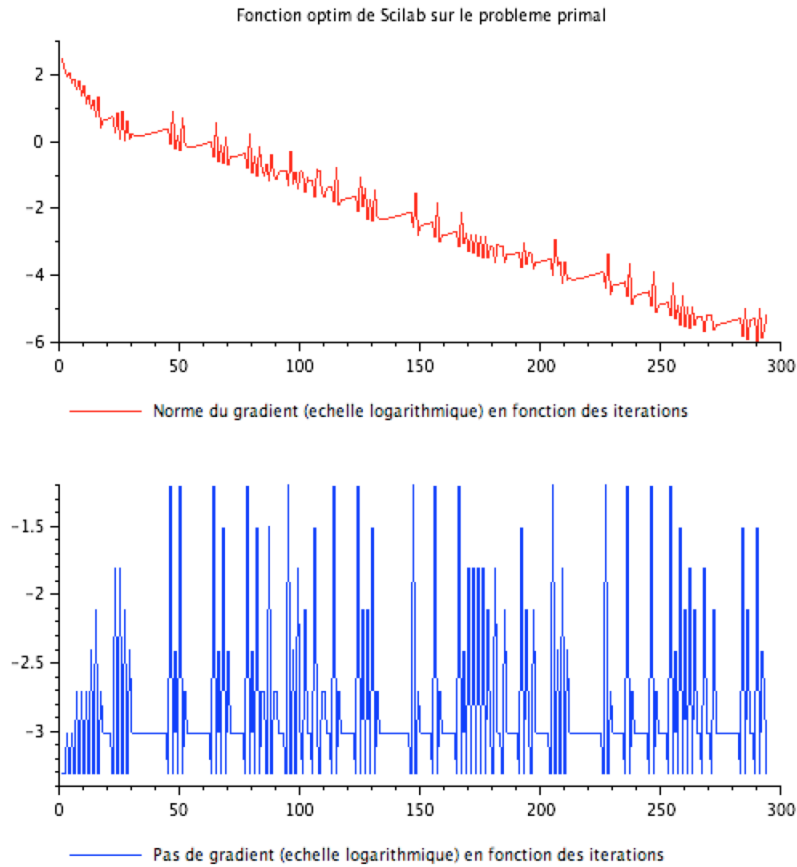
Pas de gradient 0.0005



### 1.3.2 Gradient à pas variable utilisant la recherche linéaire

Nous modifions ensuite la méthode de gradient en utilisant les conditions de Wolfe et l'algorithme de Fletcher–Lemaréchal pour calculer le pas. Il faut compiler l'algorithme **Moniteur\_Grad.sce**. Nous prenons dorénavant un pas de gradient de 1.

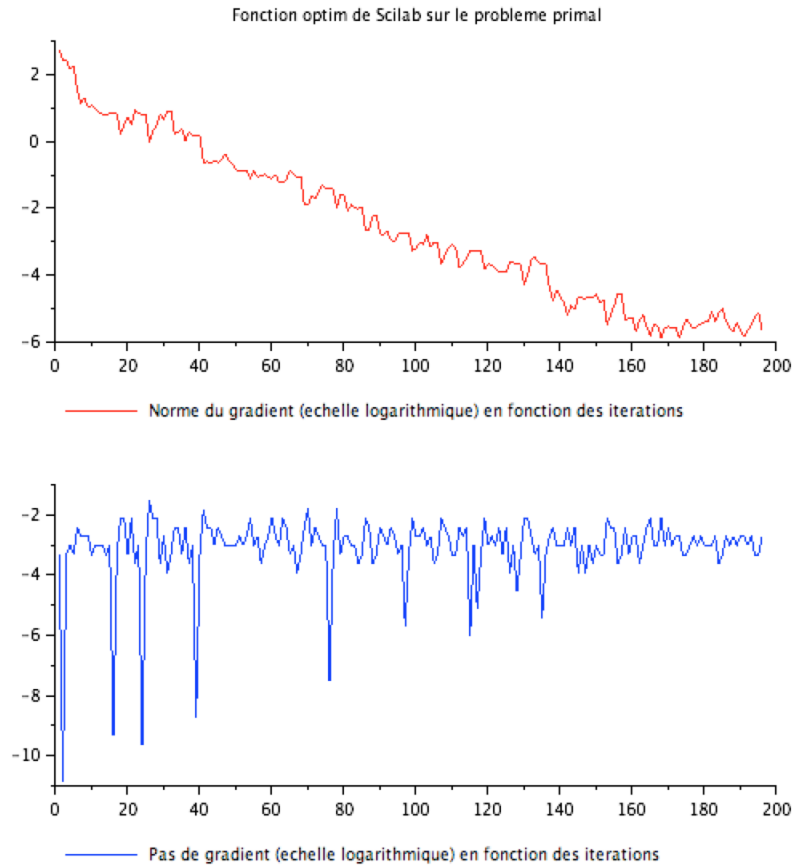
Iteration	292
Temps CPU	0.171761
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 1.3.3 Algorithme de Polak-Ribière

Ensuite, nous avons appliqué l'algorithme de Polak-Ribière qui utilise le gradient conjugué non linéaire, avec la recherche de Wolfe. Il faut compiler l'algorithme **Moniteur\_PR.sce**.

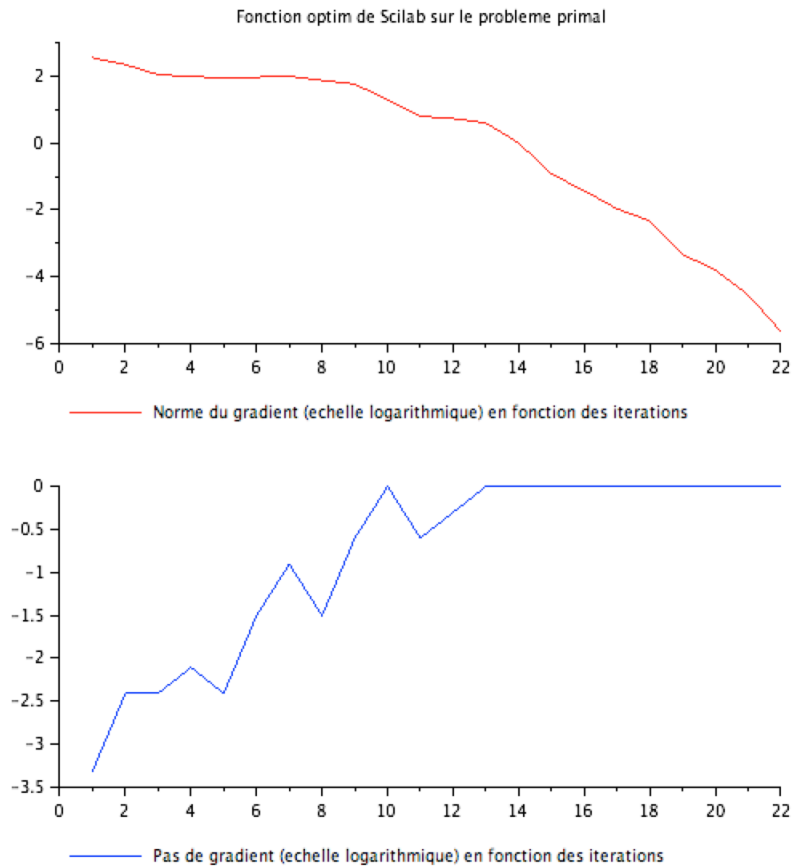
Iteration	197
Temps CPU	0.120587
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 1.3.4 Algorithme de BFGS

Nous avons appliqué l'algorithme de BFGS qui utilise une approximation de l'inverse du hessien, toujours avec la recherche de Wolfe. Il faut compiler l'algorithme **Moniteur\_BFGS.sce**.

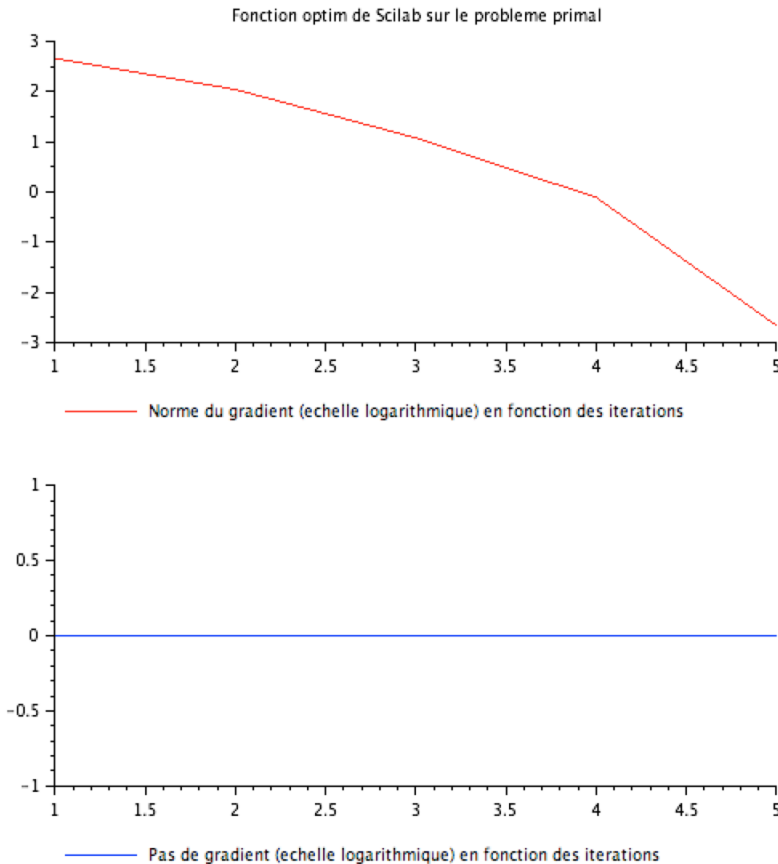
Iteration	23
Temps CPU	0.007418
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 1.3.5 Méthode de Newton avec recherche linéaire de Wolfe

Pour cette partie, nous développons la méthode de Newton qui utilise l'inverse de la hessienne. Il faut compiler l'algorithme **Moniteur\_Newton.sce**.

Iteration	6
Temps CPU	0.002211
Critere optimal	-3.734007
Norme du gradient	0.0000009



En à peine six itérations, nous obtenons la convergence de l'algorithme.

## 1.4 Conclusion

La méthode de Newton est de loin la plus efficace. Le cout de temps principal de cette méthode provenant de l'inversion de la matrice, le temps de calcul peut s'avérer plus important pour des matrices de taille supérieure. Cela est atteint car nous travaillons sur une petite instance, si il y a plus de noeuds, le calcul de l'inverse devient vite très complexe. Dans ce cas on préférera la méthode BFGS. Néanmoins, c'est la méthode de Newton dont le temps CPU est le moins important.

## 2 Problème dual

### 2.1 Calcul du gradient et de la hessienne

Le lagrangien associé au problème de minimisation suivant avec la contrainte  $A_d q - f_d = 0$

$$\min_{q \in \mathbb{R}^n} \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle \quad (10)$$

est :

$$L(\lambda, q) = \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle + \langle \lambda, A_d q - f_d \rangle \quad (11)$$

Ses conditions d'optimalités sont :

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= 0 \\ \frac{\partial L}{\partial q} &= 0 \end{aligned} \quad (12)$$

Minimisons le lagrangien en  $q$ :

$$\frac{\partial L}{\partial \lambda} = r \bullet q \bullet |q| + A_r^T p_r + A_d^T \lambda = 0 \quad (13)$$

En prenant le  $k$ -ième élément de ce vecteur.

$$\begin{aligned} q_k &= -\text{sign}(X) \sqrt{X/r_k} \\ \text{avec } X &= [A_r^T p_r + A_d^T \lambda]_k \end{aligned} \quad (14)$$

Appelons  $q_{\#}$  cette valeur qui minimise le lagrangien. La fonction duale est alors  $\phi(\lambda) = L(\lambda, q_{\#})$

$$\phi(\lambda) = L(\lambda, q_{\#}) = \frac{1}{3} \langle q_{\#}, r \bullet q_{\#} \bullet |q_{\#}| \rangle + \langle p_r, A_r q_{\#} \rangle + \langle \lambda, A_d q_{\#} - f_d \rangle \quad (15)$$

Puis en utilisant la formule de la chaîne et par construction de  $\phi$ ,

$$\nabla \phi = \frac{\partial q_{\#}}{\partial \lambda} \frac{\partial L}{\partial q}(q_{\#}, \lambda) + \frac{\partial L}{\partial \lambda} = -A_d q_{\#} - f_d \quad (16)$$

Pour calculer la hessienne, il faut encore dériver  $\nabla \phi$  par rapport à  $\lambda$ .

Il est possible de décomposer cette fonction en une composition de 3, plus simples.

$$\begin{aligned} f : \lambda &\rightarrow A_r^T p_r + A_d^T \lambda = X \\ \varphi : X &\rightarrow \sqrt{\frac{|x_k|}{r_k}} * \text{sign}(x_k) \mid_k = q_{\#} \\ g : q_{\#} &\rightarrow -A_d q_{\#} - f_d \\ \nabla \phi(\lambda) &= g \circ \varphi \circ f(\lambda) \end{aligned} \quad (17)$$

Il suffit de composer les dérivées pour arriver à la hessienne.

$$\begin{aligned} \frac{\partial \nabla \phi}{\partial \lambda} &= \frac{\partial f}{\partial \lambda} \frac{\partial \varphi}{\partial X} \frac{\partial g}{\partial q_{\#}} \\ \frac{\partial f}{\partial \lambda} &= A_d \\ \frac{\partial \varphi}{\partial x} &= \text{diag}\left(\frac{1}{2r_k \sqrt{|X|}}\right) \\ \frac{\partial g}{\partial q_{\#}} &= A_d^T \end{aligned} \quad (18)$$

$$H(\phi) = A_d \text{diag}\left(\frac{1}{2r \sqrt{|X|}}\right) A_d^T \quad (19)$$



## 2.2 Questions théoriques

On a le lagrangien du problème:

$$L(\lambda, q) = \frac{1}{3} \langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle + \langle \lambda, A_d q - f_d \rangle \quad (20)$$

ainsi que ses conditions d'optimalité:

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= 0 \\ \frac{\partial L}{\partial q} &= 0 \end{aligned} \quad (21)$$

La première est équivalente à dire qu'à l'optimal le gradient est nul. En calculant il tombe que:

$$\begin{aligned} \frac{\partial L}{\partial q} \frac{\partial q}{\partial \lambda} + A_d q - f_d &= 0 \\ \text{or } \frac{\partial L}{\partial q} &= 0 \end{aligned} \quad (22)$$

Il tombe donc que:

$$A_d q - f_d = 0 \quad (23)$$

Ainsi, l'équation d'équilibre du réseau(6) correspond à la condition  $\frac{\partial L}{\partial \lambda} = 0$ .

On se tourne vers l'équation d'équilibre du réseau (7)  $\varphi_\alpha(q_\alpha) = r_\alpha q_\alpha |q_\alpha|$ . Montrons qu'elle correspond à la condition d'optimalité  $\frac{\partial L}{\partial q} = 0$ :

$$\frac{\partial L}{\partial q} = \frac{1}{3} \frac{\partial}{\partial q} (\langle q, r \bullet q \bullet |q| \rangle + \langle p_r, A_r q \rangle + \langle \lambda, A_d q - f_d \rangle) \quad (24)$$

On développe:

$$\begin{aligned} \frac{\partial}{\partial q} (\langle q, r \bullet q \bullet |q| \rangle) &= \left[ \frac{\partial}{\partial q_k} \left( \sum_i q_i^3 r_i \right) \right]_k = 3 \text{sign}(q_k) q_k^2 |_k \\ &= 3 r \bullet q \bullet |q| \end{aligned} \quad (25)$$

On en tire que, puisqu'à l'optimal:

$$\begin{aligned} \frac{\partial L}{\partial q} &= r \bullet q \bullet |q| + A_r^T p_r + A_d^T \lambda = 0 \\ \text{donc : } r \bullet q \bullet |q| + A^T p &= 0 \end{aligned} \quad (26)$$

Ainsi, la condition  $\frac{\partial L}{\partial q} = 0$  correspond bien à l'équation d'équilibre du réseau (7)

## 2.3 Applications aux différentes méthodes

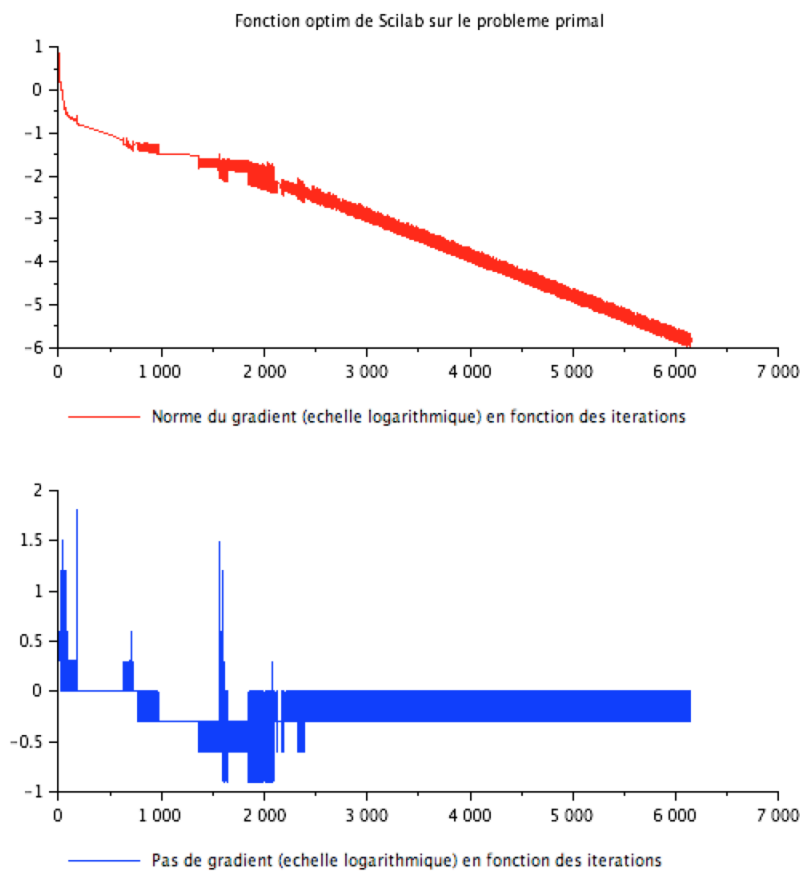
### 2.3.1 Gradient à pas fixe

La méthode du gradient à pas fixe ne convient pas pour la fonction duale.

### 2.3.2 Gradient à pas variable utilisant la recherche linéaire

Il faut compiler l'algorithme `MoniteurD_Grad.sce`.

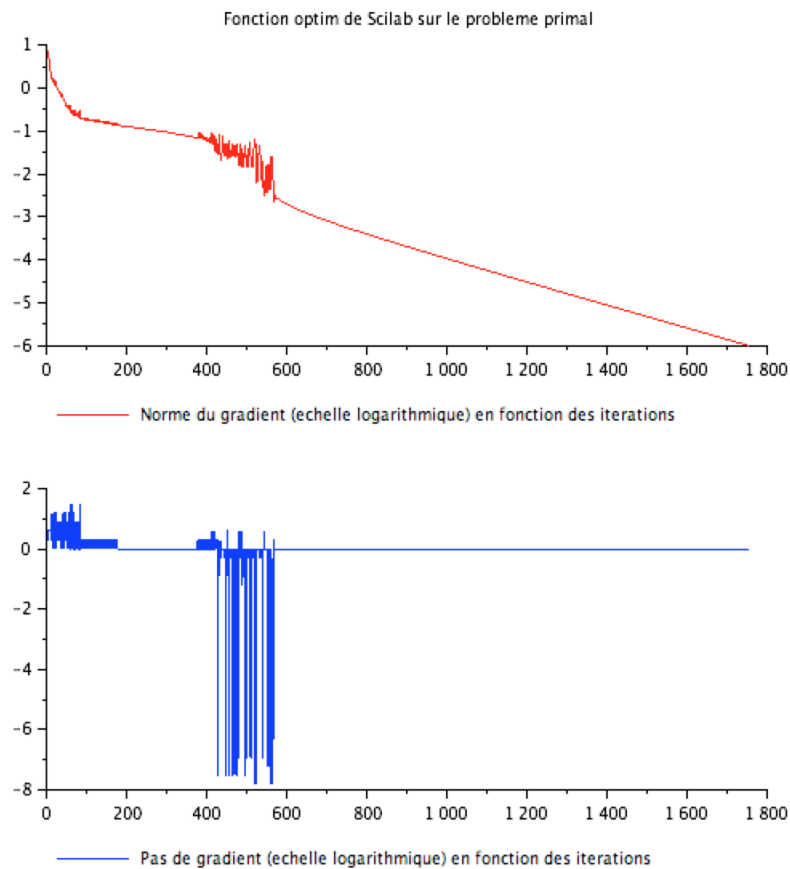
Iteration	6146
Temps CPU	1.185736
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 2.3.3 Algorithme de Polak-Ribière

Il faut compiler l'algorithme `MoniteurD_PR.sce`.

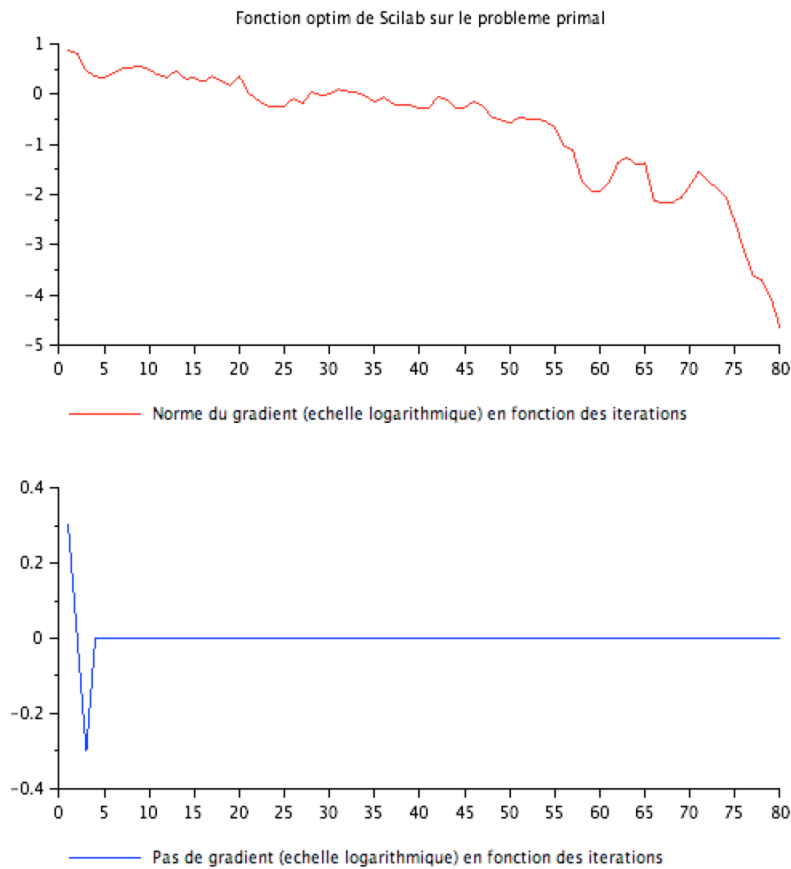
Iteration	1754
Temps CPU	0.350287
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 2.3.4 Algorithme de BFGS

Il faut compiler l'algorithme **MoniteurD\_BFGS.sce**.

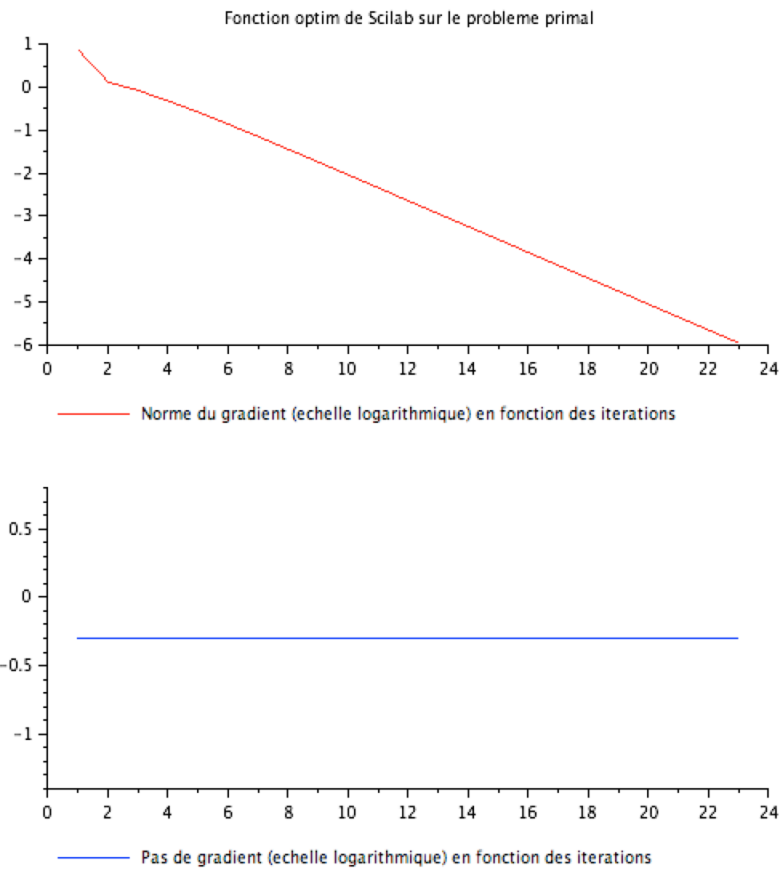
Iteration	81
Temps CPU	0.043883
Critere optimal	-3.734007
Norme du gradient	0.0000009



### 2.3.5 Méthode de Newton avec recherche linéaire de Wolfe

Il faut compiler l'algorithme `MoniteurD_Newton.sce`.

Iteration	24
Temps CPU	0.011409
Critere optimal	-3.734007
Norme du gradient	0.0000009



## 2.4 Conclusion

Comme pour l'approche primale, la méthode de Newton fournit les meilleurs résultats en temps CPU et en nombres d'itérations. Par ailleurs, la méthode de la fonction duale n'apporte pas de meilleurs résultats que l'approche primale. En effet, chaque algorithme utilisé avec la fonction duale sous performe par rapport à son utilisation avec la fonction primale. La remarque faite concernant l'inversion de la hesienne dans le cas du problème primal est toujours valable.