

# Proximal-Point Algorithm for Minimizing Quadratic Functions

Guillaume Vige

► **To cite this version:**

Guillaume Vige. Proximal-Point Algorithm for Minimizing Quadratic Functions. [Research Report] RR-2610, INRIA. 1995. <inria-00074075>

**HAL Id: inria-00074075**

**<https://hal.inria.fr/inria-00074075>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***Proximal-point algorithm for minimizing  
quadratic functions***

Guillaume VIGE

**N° 2610**

Juillet 1995

PROGRAMME 5

Traitement du signal,  
automatique  
et productique



***Rapport  
de recherche***

**1995**





## Proximal-point algorithm for minimizing quadratic functions

Guillaume VIGE

Programme 5 — Traitement du signal, automatique et productique  
Projet Promath

Rapport de recherche n° 2610 — Juillet 1995 — 18 pages

**Abstract:** The proximal-point algorithm is studied in the framework of the minimization of a quadratic function. In this context it can also be used to compute the least eigenvalue of the hessian and an associated eigenvector. The speed of convergence is analysed and a connection with the power method is pointed out. A modified algorithm with quadratic convergence is studied. Numerical tests assess the theoretical results.

*(Résumé : tsvp)*

## **Etude de l'algorithme du point proximal appliqué à la minimisation d'une fonction quadratique**

**Résumé :** L'algorithme du point proximal est étudié dans le cadre de la minimisation d'une fonction quadratique. On montre qu'il permet d'obtenir la plus petite valeur propre du hessien de la fonction et un vecteur propre associé. La vitesse de convergence est analysée, on met en évidence un lien avec la méthode de la puissance itérée. On illustre les résultats théoriques sur les vitesses de convergence par des tests numériques.

**Mots-clé :** Proximal point algorithm

## 1 Introduction

L'algorithme du point proximal (APP) a été introduit par Martinet en 1970 [6] et d'importants résultats ont été publiés par Rockafellar en 1976 [8]. Il s'agit d'un algorithme général pour trouver un zéro d'un opérateur monotone maximal  $T$  comme limite de la suite  $\{x_k\}$  produite par l'itération

$$x_{k+1} = (I + c_k T)^{-1}(x_k).$$

Le sous-différentiel d'une fonction  $f$ , convexe, propre, semi-continue inférieurement est un opérateur monotone maximal. Pour minimiser une fonction en trouvant un zéro de son sous-différentiel, on peut donc utiliser l'APP. On peut alors montrer que

$$x_{k+1} = \operatorname{Argmin}_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}. \quad (1)$$

Pour une fonction  $f$  quelconque, il n'est généralement pas possible de calculer le minimum exact dans (1). Il n'est pas non plus souhaitable de réaliser ce calcul avec une grande précision puisque seule la limite de  $\{x_k\}$  est intéressante. C'est pourquoi Rockafellar [8], Auslender [1] et Correa-Lemaréchal [3] ont étudié la convergence de l'APP lorsque ce minimum est seulement calculé de manière approchée.

Cependant un calcul exact est possible (au moins théoriquement) lorsque  $f$  est une fonction polyédrique ou une fonction quadratique. C'est ce dernier cas que nous étudions ici. Soit  $f$  une fonction quadratique définie sur  $\mathbb{R}^n$  de la forme

$$f(x) = \frac{1}{2} \langle x, Qx \rangle - \langle b, x \rangle, \quad (2)$$

avec  $Q$  symétrique, semi-définie positive.

Comme  $f$  est différentiable, le sous-différentiel de  $f$  est réduit à son gradient et l'algorithme du point proximal prend la forme

$$x_{k+1} = (I + c_k \nabla f)^{-1} x_k$$

donc

$$x_{k+1} + c_k \nabla f(x_{k+1}) = x_k,$$

c'est-à-dire

$$(x_{k+1} + c_k(Qx_{k+1} - b)) = x_k,$$

soit

$$x_{k+1} = (I + c_k Q)^{-1}(c_k b + x_k).$$

**Définition 1** Une suite  $\{x_k\}$  converge linéairement vers  $x^*$ , avec un taux  $\alpha$ , s'il existe  $\alpha \in ]0, 1[$  tel que

$$\limsup_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \alpha \quad (3)$$

La convergence est quadratique si et seulement si

$$\limsup_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} < +\infty \quad (4)$$

Rockafellar a établi que si  $T^{-1}$  est Lipschitz-continue en 0 (nous ne précisons pas cette notion ici), l'APP converge linéairement [8, Théorème 2]. Il a également montré, [9, Proposition 3] que lorsque  $T = \partial f$ ,  $T$  est Lipschitz-continue en 0 si et seulement si les deux conditions suivantes sont satisfaites:

- $f$  a un minimum unique  $x^*$ ,
- $\exists \lambda > 0, \exists \varepsilon > 0$  tels que:  
 $f(x) \geq f(x^*) + \lambda \|x - x^*\|^2 \quad \forall x \text{ tel que } \|x - x^*\| \leq \varepsilon.$

Cette hypothèse est satisfaite pour une fonction quadratique dont le hessien  $Q$  est défini positif avec  $x^* = Q^{-1}b$  et  $\lambda$  égal à la plus petite valeur propre de  $Q$ .

Si  $Q$  est seulement semi-défini positif et  $\text{Argmin } f \neq \emptyset$ , un résultat de Luque [5, Théorème 2.1] entraîne de nouveau la convergence linéaire de  $\{x_k\}$ .

## 2 Modification de l'APP pour le calcul d'éléments propres

Considérons maintenant l'APP dans lequel  $\{c_k\}$  est une suite constante (quel que soit  $k$ ,  $c_k=c$ ). Alors il suffit d'inverser  $(I + cQ)$  une seule fois au départ et l'évaluation de l'application proximale se réduit à une multiplication par la matrice  $M = (I + cQ)^{-1}$ . Pour une plus grande efficacité numérique, on peut remplacer l'inversion matricielle par une factorisation (Cholesky, QR, ...).

Dans l'algorithme suivant, on introduit deux nouvelles suites  $\{w_k\}$  et  $\{\gamma_k\}$  obtenues à partir de  $\{x_k\}$ ;  $b \in \mathbb{R}^n$  a été défini en (2) et  $\|\cdot\|_2$  désigne la norme euclidienne dans  $\mathbb{R}^n$ .

Algorithme du Point Proximal (APP):

- Choisir  $x_0 \in \mathbb{R}^n$ , poser  $M = (I + cQ)^{-1}$  et  $k = 0$

- A chaque itération faire:

$$x_{k+1} = M(cb + x_k)$$

$$y_{k+1} = x_{k+1} - x_k$$

$$w_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|_2}$$

$$\gamma_{k+1} = \frac{\|My_{k+1}\|_2}{\|y_{k+1}\|_2}$$

Incrémenter  $k$ . ■

Comme on peut déjà le remarquer, dans le cas  $c = 1$  et  $b = 0$ , l'APP n'est rien d'autre que l'algorithme de la puissance itérée [4, 7], pour le calcul d'éléments propres.

Soit  $M$  une matrice symétrique réelle de dimension  $n$ ,  $\lambda_1, \dots, \lambda_m$  ses  $m$  valeurs propres distinctes telles que  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_m|$ . Nous noterons  $E_i$  l'espace propre associé à  $\lambda_i$  et  $v_i$  un vecteur propre non nul de  $E_i$ .

La convergence linéaire de la méthode de la puissance itérée avec un taux égal à  $|\frac{\lambda_2}{\lambda_1}|$  est un résultat classique d'analyse matricielle. Le lemme suivant montre que dans le cas d'une matrice symétrique, ce taux est ramené à  $(\frac{\lambda_2}{\lambda_1})^2$  si l'on utilise la norme euclidienne.

**Lemme 1** Soit  $x_0$  non orthogonal à  $E_1$  et  $x_k = M^k x_0$ ; alors la suite  $\delta_k = \frac{\|x_{k+1}\|_2}{\|x_k\|_2}$  converge vers  $|\lambda_1|$ .

Si, de plus,  $x_0$  n'est pas orthogonal à  $E_2$ , la convergence est linéaire avec un taux égal à  $(\lambda_2/\lambda_1)^2$ .

### Démonstration

Soit  $v_1, \dots, v_m$  une famille de vecteurs propres orthonormés tels que  $x_0 = \beta_1 v_1 + \dots + \beta_m v_m$ , alors

$$M^k x_0 = \beta_1 \lambda_1^k v_1 + \dots + \beta_m \lambda_m^k v_m.$$

Les vecteurs  $v_i$  étant orthonormés, on a d'après Pythagore:

$$\|M^k x_0\|_2^2 = \sum_{i=1}^m \beta_i^2 \lambda_i^{2k}; \quad (5)$$

par hypothèse  $\beta_1 \neq 0$ , donc

$$\|M^k x_0\|_2^2 = \beta_1^2 \lambda_1^{2k} \left( 1 + \sum_{i=2}^m \left( \frac{\beta_i}{\beta_1} \right)^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \right).$$



Si  $x_0$  n'est pas orthogonal à  $E_2$ , alors  $\beta_2 \neq 0$ . Soient  $\omega = \left(\frac{\beta_2}{\beta_1}\right)^2$  et  $\tau_k = \left(\frac{\lambda_2}{\lambda_1}\right)^{2k}$ ; lorsque  $k$  tend vers  $+\infty$ ,  $\tau_k$  tend vers 0 donc

$$\|M^k x_0\|_2^2 \sim \beta_1^2 \lambda_1^{2k} (1 + \omega \tau_k),$$

donc

$$\delta_k^2 \sim \lambda_1^2 \frac{1 + \omega \left(\frac{\lambda_2}{\lambda_1}\right)^2 \tau_k}{1 + \omega \tau_k} \sim \lambda_1^2 \left(1 + \omega \left(\left(\frac{\lambda_2}{\lambda_1}\right)^2 - 1\right) \tau_k\right),$$

d'où

$$\delta_k \sim |\lambda_1| \left(1 + \frac{\omega}{2} \left(\left(\frac{\lambda_2}{\lambda_1}\right)^2 - 1\right) x\right),$$

et par conséquent

$$\delta_k - |\lambda_1| = O(x) = O\left(\left(\frac{\lambda_2}{\lambda_1}\right)^2\right)^k.$$

■

**Remarque 1** L'égalité (5) est due d'une part au fait que les vecteurs  $v_i$  peuvent être choisis orthogonaux (car  $M$  est symétrique) et d'autre part à l'utilisation de la norme euclidienne.

Il s'avère que pour les autres normes usuelles ( $\|\cdot\|_1, \|\cdot\|_\infty$ ), le taux de convergence des suites  $\{\delta_k\}$  et  $\{\gamma_k\}$  ne vaut que  $|\frac{\lambda_2}{\lambda_1}|$  comme l'illustrent le test 2 et la figure 3.

**Remarque 2** Si  $x_0$  est orthogonal à  $E_2$ , on montre de la même façon que le taux de convergence est égal à  $(\lambda_{i_0}/\lambda_1)^2$  où  $i_0$  est le plus petit indice  $i$  tel que  $x_0$  n'est pas orthogonal à  $E_i$ .

Si  $x_0$  est orthogonal à  $E_i$  pour tout  $i \geq 2$  alors  $x_0$  est un vecteur propre associée à  $\lambda_1$ ,  $\delta_0 = |\lambda_1|$  donc l'algorithme converge en une itération.

**Théorème 1** Soit  $f$  une fonction quadratique dont le hessien  $Q$  est semi-défini positif,  $\alpha_1$  la plus petite valeur propre non nulle de  $Q$  et  $\{x_k\}$  une suite de points générés par l'APP avec pour tout  $k$ ,  $c_k = c$ . On suppose que  $x_0$  n'est pas orthogonal à l'espace propre associé à  $\alpha_1$  alors:

$$\lim_{k \rightarrow +\infty} \gamma_k = \frac{1}{1 + c\alpha_1}$$

et

$$\lim_{k \rightarrow +\infty} w_k = v_1,$$

où  $v_1$  est un vecteur propre associé à  $\alpha_1$ . Les suites  $\{\gamma_k\}$  et  $\{w_k\}$  convergent linéairement avec un taux respectivement égal à  $\left(\frac{1+c\alpha_1}{1+c\alpha_{p+1}}\right)^2$  et  $\frac{1+c\alpha_1}{1+c\alpha_{p+1}}$  où  $p$  est la multiplicité de la valeur propre  $\alpha_1$ .

Si  $\text{Argmin } f \neq \emptyset$  alors  $\{x_k\}$  converge linéairement vers  $x^* = \text{proj}_{\text{Argmin } f} x_0$  avec un taux égal à  $\frac{1}{1+c\alpha_1}$ .

### Démonstration

Sachant que  $y_{k+1} = x_{k+1} - x_k$ , on obtient immédiatement par différence

$$y_{k+1} = (I + cQ)^{-1}(y_k), \quad (6)$$

ce qui n'est rien d'autre que la méthode de la puissance itérée avec la matrice  $(I + cQ)^{-1}$ .

Soient  $\alpha_1, \dots, \alpha_m$  les valeurs propres de  $Q$  et  $E_i$  l'espace propre associé à  $\alpha_i$ ; alors  $E_i$  est aussi un espace propre de  $(I + cQ)^{-1}$  associé à la valeur propre  $\frac{1}{1+c\alpha_i}$ . D'après le théorème de convergence de la méthode de la puissance itérée (voir par exemple [2, Théorème 6.4-1]),  $\{w_k\}$  converge vers  $v_1 \in E_1$  avec un taux égal à  $\frac{1+c\alpha_1}{1+c\alpha_{p+1}}$  et d'après le Lemme 1,  $\{\delta_k\}$  et  $\{\gamma_k\}$  convergent vers  $\frac{1}{1+c\alpha_1}$  avec un taux égal à  $\left(\frac{1+c\alpha_1}{1+c\alpha_{p+1}}\right)^2$ .

Si  $\text{Argmin } f \neq \emptyset$  alors la convergence linéaire de  $\{x_k\}$  vers  $x^* \in \text{Argmin } f$  découle directement des résultats de Rockafellar [8, Théorème 2]. Montrons que  $x^*$  est la projection de  $x_0$  sur  $\text{Argmin } f$ .

Plaçons nous dorénavant dans une base  $E$  de vecteurs propres de  $Q$ .

$$\text{Argmin } f = \{x : \nabla f(x) = 0\} = \{x : \alpha_i x^i = b^i\},$$

donc  $\alpha_i = 0$  implique  $b_i = 0$  puisque par hypothèse  $\text{Argmin } f \neq \emptyset$ .

Dans la base  $E$ , pour chaque composante  $x_{k+1}^i$  de  $x_{k+1}$ , l'itération (??) devient

$$x_{k+1}^i = \frac{1}{1+c\alpha_i}(cb_i + x_k^i).$$

Si  $\alpha_i = 0$ , on a alors  $x_{k+1}^i = x_k^i = x_0^i$  donc  $x_i^* = x_0^i$ , c'est-à-dire

$$\text{proj}_{\ker Q}(x^* - x_0) = 0.$$

Comme  $x^* \in \text{Argmin } f$ , on a  $\text{Argmin } f = x^* + \ker Q$  et par conséquent  $x^*$  est la projection de  $x_0$  sur  $\text{Argmin } f$ . ■

**Remarque 3** On peut interpréter ce résultat géométriquement en disant que la trajectoire générée par l'APP est tangente en  $x^* = \lim x_k$  à un vecteur propre associé à la plus petite valeur propre non nulle de  $Q$  (voir Figure 1).

**Remarque 4** En posant  $b = 0$  et  $c_k$  constant, l'APP se réduit à une inversion (ou une factorisation) de la matrice  $(I + cQ)$  au départ, puis à un produit matrice-vecteur à chaque itération.

Exemple:

Soit  $Q$  la matrice  $\begin{pmatrix} \frac{3}{10} & \frac{2}{10} \\ \frac{2}{10} & \frac{3}{10} \end{pmatrix}$ , ses valeurs propres sont  $\alpha_1 = \frac{1}{10}$  et  $\alpha_2 = \frac{1}{2}$  associées respectivement aux vecteurs propres  $v_1 = (-1, 1)$  et  $v_2 = (1, 1)$ .

La Figure 1 représente les 30 premières itérations de l'APP appliqué à la minimisation de la fonction  $f$  définie sur  $\mathbb{R}^2$  par  $f(x) = \frac{1}{2}\langle x, Qx \rangle$  avec  $x_0 = (\frac{7}{10}, \frac{1}{2})$  et  $c = 1$ . Un vecteur colinéaire à  $v_1$  a été dessiné en  $x^* = (0, 0)$  pour illustrer le fait que la trajectoire devient tangente à  $v_1$  au voisinage de  $x^*$ .

Comme  $\alpha_1$  est petit, la convergence est relativement lente puisque le taux est égal à  $\frac{10}{11}$ .

La remarque 4 permet de construire un algorithme proximal amélioré pour calculer la plus petite valeur propre d'une matrice  $Q$  symétrique, semi-définie positive ainsi qu'un vecteur propre associé.

APPQ:

- Choisir  $\tilde{x}_0 \in \mathbb{R}^n$ , poser  $M_0 = (I + cQ)^{-1}$  et  $k = 0$

- A chaque itération faire:

$$M_{k+1} = (M_k)^2$$

$$\tilde{x}_{k+1} = M_{k+1} \tilde{x}_0$$

$$\tilde{w}_{k+1} = \frac{\tilde{x}_{k+1}}{\|\tilde{x}_{k+1}\|}$$

$$\tilde{\gamma}_{k+1} = \frac{\|M_0 \tilde{x}_{k+1}\|}{\|\tilde{x}_{k+1}\|}$$

Incrémenter  $k$ . ■

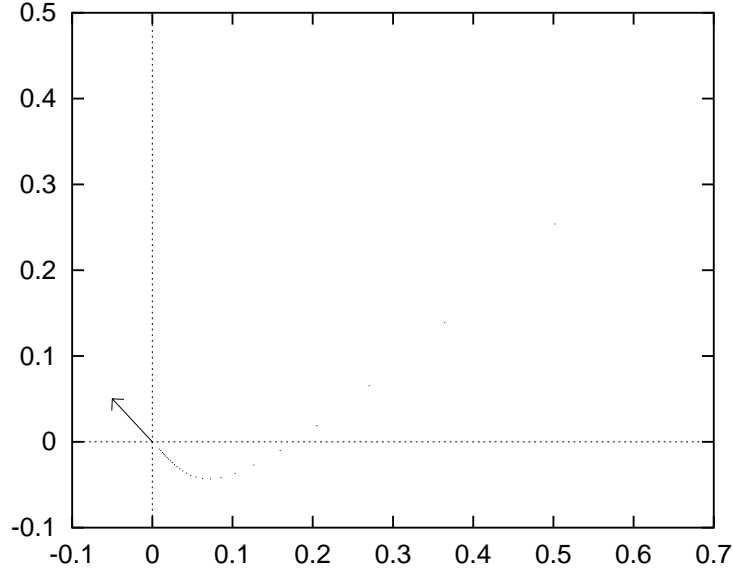


Figure 1: La trajectoire de l'APP est tangente au vecteur  $v_1$  en  $x^*$

**Remarque 5** Si  $\tilde{x}_0 = x_0$ , alors les suites  $\{\tilde{x}_k\}$ ,  $\{\tilde{w}_k\}$  et  $\{\tilde{\gamma}_k\}$  générées par l'APPQ sont des sous-suites respectives de  $\{x_k\}$ ,  $\{w_k\}$  et  $\{\gamma_k\}$  obtenues avec l'APP. Plus précisément:

$$\tilde{x}_k = x_{2k} \quad \tilde{w}_k = w_{2k} \quad \tilde{\gamma}_k = \gamma_{2k}.$$

**Théorème 2** Sous les hypothèses du Théorème 1, les suites  $\{\tilde{x}_k\}$ ,  $\{\tilde{\gamma}_k\}$  et  $\{\tilde{w}_k\}$  générées par l'APPQ convergent respectivement vers  $x^* = \text{proj}_{\ker Q} x_0$ ,  $\frac{1}{1+c\alpha_1}$  et  $v_1$  et la convergence est quadratique.

### Démonstration

La convergence des suites  $\{\tilde{x}_k\}$ ,  $\{\tilde{\gamma}_k\}$  et  $\{\tilde{w}_k\}$  ainsi que leurs limites respectives sont des conséquences immédiates de la remarque 5 et du Théorème 1.

Comme cela est indiqué dans [4, Théorème 2, p 594] pour l'algorithme de la puissance itérée, non seulement les suites  $\{x_k\}$ ,  $\{w_k\}$  et  $\{\gamma_k\}$  convergent linéairement mais la limite supérieure dans (3) est en fait une limite.

On peut donc écrire

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \gamma = \frac{1}{1 + c\alpha_1} \quad (7)$$

donc

$$\|x_{k+1} - x^*\| \sim \gamma \|x_k - x^*\|.$$

En posant  $\omega = \|x_0 - x^*\|$ , on obtient par récurrence

$$\|x_k - x^*\| \sim \omega \gamma^k,$$

et par conséquent

$$\|\tilde{x}_{k+1} - x^*\| = \|x_{2^{k+1}} - x^*\| \sim \omega \gamma^{2^{k+1}}$$

et

$$\|\tilde{x}_k - x^*\|^2 \sim (\omega \gamma^{2^k})^2 = \omega^2 \gamma^{2^{k+1}}$$

de sorte que

$$\frac{\|\tilde{x}_{k+1} - x^*\|}{\|\tilde{x}_k - x^*\|^2} \sim \frac{1}{\omega}$$

d'où la convergence quadratique de  $\tilde{x}_k$ .

Le même raisonnement peut être utilisé pour montrer la convergence quadratique de  $\tilde{w}_k$  et  $\tilde{\gamma}_k$ . ■

**Remarque 6** *On peut bien sûr améliorer l'APPQ en posant  $\tilde{x}_{k+1} = M_{k+1} \tilde{x}_k$ , de sorte que  $\tilde{x}_{k+1} = M_0^{2^{k+1}} M_0^{2^k} \dots M_0 \tilde{x}_0 = M_0^{2^{k+2}} M_0^{-1} x_0$  ce qui permet de gagner "asymptotiquement" une itération. La convergence reste donc quadratique.*

**Remarque 7** *Comme dans la méthode de la puissance itérée, le Théorème 2 reste vrai si  $Q$  n'est pas symétrique et si  $Q$  n'est pas semi-définie positive ( $\{\tilde{x}_k\}$  peut ne pas converger et  $\{\tilde{\gamma}_k\}$  converge vers la plus petite des valeurs absolues des  $\frac{1}{1+c\alpha_i}$ ). Dans ce cas il est toutefois difficile de continuer à parler "d'algorithme proximal".*

### 3 Analyse de la complexité

Il est clair que l'APP et l'APPQ appliqués à la minimisation d'une fonction quadratique n'ont d'intérêt que parce qu'ils permettent également le calcul de la plus petite valeur propre du hessien et d'un vecteur propre associé.

Comme nous l'avons vu dans le paragraphe précédent, les suites générées par l'APPQ convergent beaucoup plus rapidement que les suites obtenues avec l'APP. Cependant, chaque itération de l'APPQ nécessite la multiplication de deux matrices tandis qu'une itération de l'APP ne nécessite que la multiplication d'une matrice par un vecteur.

Nous comparons ici le coût des deux algorithmes en terme d'opérations élémentaires sur les réels en fonction de  $n$  et de  $m$ , où  $n$  est la dimension de  $Q$  et  $m$  le nombre de chiffres significatifs souhaités pour la valeur propre  $\alpha_1$ .

#### 3.1 Coût de l'APP

Si l'on ne s'intéresse qu'au calcul de  $\alpha_1$ , on peut choisir  $b = 0$  et l'itération de l'APP est alors constituée des deux étapes:

$$x_{k+1} = M_0 x_k$$

et

$$\gamma_{k+1} = \frac{\|x_{k+1}\|}{\|x_k\|}$$

donc elle nécessite un produit matrice-vecteur ( $n(2n - 1)$  opérations), le calcul de la norme de  $x_{k+1}$  ( $2n - 1$  opérations) et une division soit  $2n^2 + n = O(n^2)$  opérations élémentaires.

D'après (7), le gain de précision par itération est asymptotiquement constant. Autrement dit, le nombre de chiffres significatifs de  $\gamma_k$  est une fonction asymptotiquement affine de  $k$ , par conséquent, le nombre d'itérations de l'APP nécessaires pour gagner  $m$  chiffres significatifs vaut  $O(m)$ . L'APP a donc un coût égal à  $O(mn^2)$ .

#### 3.2 Coût de l'APPQ

L'essentiel du coût d'une itération de l'APPQ est dû à la multiplication matrice-matrice soit environ  $2n^3$  opérations élémentaires.

La convergence quadratique correspond au doublement du nombre de chiffres significatifs à chaque itération (asymptotiquement). A une constante près, le nombre d'itérations nécessaires pour gagner  $m$  chiffres significatifs vaut donc  $\log(m)/\log(2)$ ,

soit  $O(\log(m))$ .

Le coût de l'APPQ est donc  $O(n^3 \log(m))$ .

### 3.3 Conclusion

Récapitulons les résultats précédents dans un tableau comparatif.

	APP	APPQ
Formule de mise à jour	$x_{k+1} = M_0 x_k$	$M_{k+1} = (M_k)^2$ $\tilde{x}_{k+1} = M_{k+1} \tilde{x}_k$
Coût d'une itération	$O(n^2)$	$O(n^3)$
Vitesse de convergence	linéaire	quadratique
Coût pour gagner $m$ chiffres significatifs	$O(mn^2)$	$O(n^3 \log m)$

On constate que l'APPQ est d'autant meilleur (par rapport à l'APP) que  $n$  est petit et que  $m$  est grand. Pour une précision  $m$  donnée, l'APPQ sera préférable lorsque  $n^3 \log m$  sera au plus de l'ordre de  $n^2 m$  c'est-à-dire lorsque  $n$  sera au plus de l'ordre de  $m / \log m$ . En pratique, cet ordre de grandeur est impossible à déterminer a priori, puisqu'il dépend notamment du taux de convergence de l'APP et par conséquent des valeurs propres de  $Q$ .

## 4 Comparaison avec les méthodes classiques

Les descriptions précises des méthodes suivantes se trouvent dans la plupart des ouvrages d'analyse matricielle et en particulier dans [2],[4] et [7].

### 4.1 Méthode de la puissance itérée

Elle consiste à construire la suite  $x_{k+1} = M x_k$  avec  $M = Q^{-1}$  (puisque l'on cherche la plus petite valeur propre de  $Q$ )

La suite  $\gamma_k$  correspondante converge linéairement avec un taux égal à  $\left(\frac{\alpha_1}{\alpha_{p+1}}\right)^2$  donc meilleur que celui de l'APP:  $\left(\frac{1+c\alpha_1}{1+c\alpha_{p+1}}\right)^2$ .

### 4.2 Méthode de la puissance inverse

On pose  $x_{k+1} = M^{-1} x_k$  où  $M = (Q - cI)$  et  $c$  est choisi aussi proche que possible de la valeur propre recherchée.

En supposant  $0 < c < \alpha_1$  de sorte que  $M$  reste définie positive, la convergence de  $\gamma_k$  est linéaire avec un taux égal à  $\left(\frac{\alpha_1 - c}{\alpha_{p+1} - c}\right)^2$ . Ce taux est d'autant meilleur que  $c$  est proche de  $\alpha_1$ , mais  $M$  est d'autant plus mal conditionnée.

Le calcul de  $x_{k+1}$  qui nécessite l'inversion de  $M$  peut donc être entaché d'une erreur importante. Cependant, on montre, voir par exemple [7, p 65-67], que cette erreur est presque entièrement dans la direction du vecteur propre  $v_1$ , de sorte que le mauvais conditionnement n'empêche pas la convergence vers l'élément propre.

### 4.3 L'itération inverse de Rayleigh

On choisit  $x_{k+1} = M_k x_k$ , avec  $M_k = (Q - c_k I)^{-1}$  où  $c_k = \langle Q \frac{x_k}{\|x_k\|}, \frac{x_k}{\|x_k\|} \rangle$  est appelé coefficient de Rayleigh.

La convergence est cubique donc très rapide et contrairement à la méthode de la puissance inverse, elle ne nécessite pas la connaissance à priori d'une estimation de  $\alpha_1$ . Par contre, la limite est une valeur propre quelconque (pas nécessairement la plus petite) et de plus le coût peut être important puisqu'il faut calculer à chaque itération l'inverse de  $(Q - c_k I)$ .

## 5 Tests numériques

Les tests portent seulement sur les suites  $\{\gamma_k\}$  et  $\{\tilde{\gamma}_k\}$ , générées respectivement par l'APP et l'APPQ, car leur comportement est caractéristique de chacun des algorithmes.

### 5.1 Test 1

Les matrices testées sont de la forme  $Q = H^t H + \lambda I$  où  $H$  est une matrice carrée aléatoire de dimension 100,  $I$  représente la matrice identité et  $\lambda$  est un réel choisi pour que les valeurs propres de  $Q$  soient voisines. Ce test a été réalisé avec le logiciel Matlab sur une station DEC alpha.

La Figure 2 représente  $\gamma_k - \gamma^*$  en fonction de  $k$  avec une échelle logarithmique. On observe que la courbe admet une droite pour asymptote, ce qui est caractéristique du comportement "asymptotiquement géométrique" de  $|\gamma_k - \gamma^*|$ . En effet:

$$|\gamma_k - \gamma^*| \sim q^k |\gamma_0 - \gamma^*|$$

implique

$$\log |\gamma_k - \gamma^*| \sim \log |\gamma_0 - \gamma^*| + k \log q$$



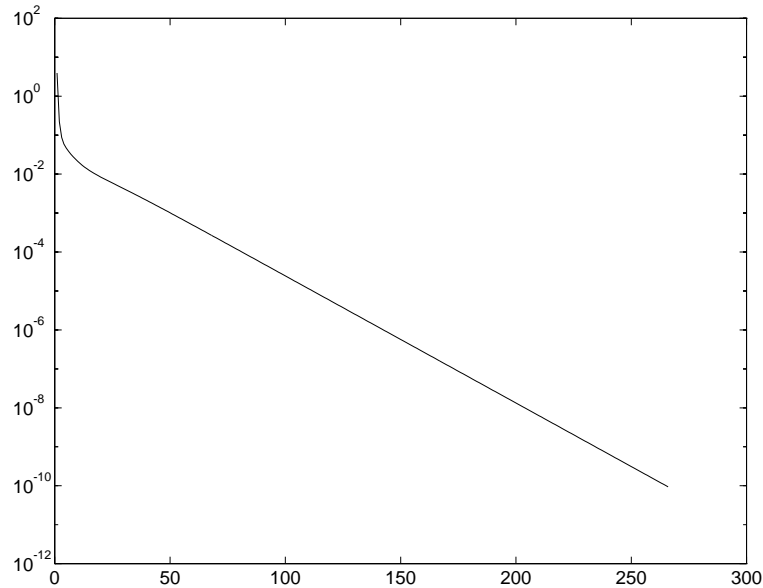


Figure 2:  $\gamma_k - \gamma^*$  en fonction de  $k$ . (Test 1)

## 5.2 Test 2

Dans ce test, les matrices sont du même type qu'au test 1, mais de dimension 10. On compare la convergence de  $\gamma_k$  pour différentes normes. La Figure 3 montre que dans les trois cas, la courbe admet une droite pour asymptote mais on remarque que la pente de l'asymptote est plus grande (en valeur absolue) pour la norme euclidienne que pour les normes  $l_1$  et  $l_\infty$ . Cela se traduit évidemment par une convergence plus rapide (20 itérations au lieu de 34 et 37 pour une précision de  $10^{-8}$ ).

## 5.3 Test 3

Pour les matrices du test 1 et l'APPQ, la Figure 4 représente  $\tilde{\gamma}_k - \gamma^*$  en fonction de  $k$  avec une échelle logarithmique. Elle met en évidence le doublement du nombre de chiffres significatifs à chaque itération (une neuvième itération dépasserait la

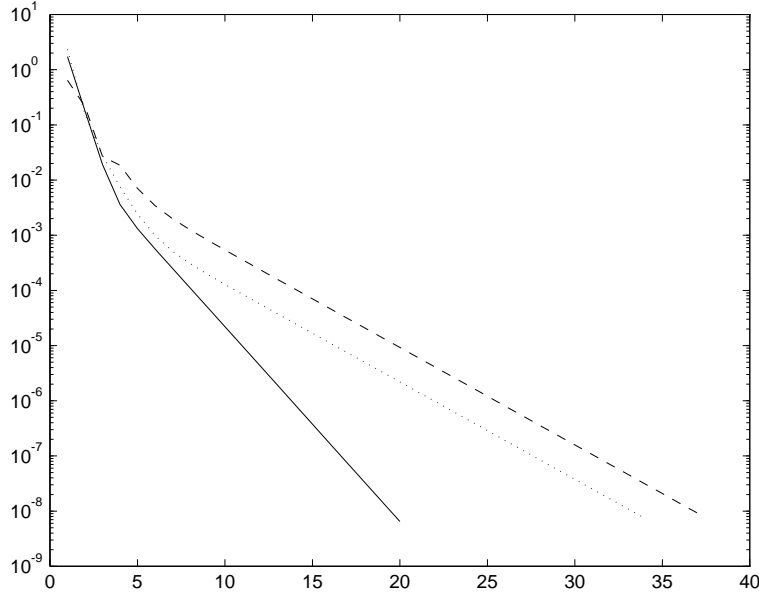


Figure 3:  $\gamma_k - \gamma^*$  en fonction de  $k$  pour  $\|\cdot\|_2$ ,  $\|\cdot\|_1$  et  $\|\cdot\|_\infty$  (Test 2)

précision de la machine).

Le faible nombre d'itérations (8 au lieu de 267 pour l'APP) ne doit pas faire oublier que le coût d'une itération est ici beaucoup plus élevé. Le temps CPU nécessaire au calcul de  $\alpha_1$  à  $10^{-12}$  près, qui est à peu près identique pour les deux algorithmes, n'est pas significatif car dans les deux cas, c'est le coût de l'inversion matricielle du départ qui domine.

#### 5.4 Test 4

Soit  $Q$  la matrice de Hilbert d'ordre 9, c'est-à-dire la matrice dont les éléments sont  $Q_{ij} = \frac{1}{i+j-1}$ . Les valeurs propres de  $Q$  valent approximativement:

1.7, 3.2e-1, 3.1 e-2, 2.0 e-3, 8.7 e-5, 2.7 e-6, 5.4 e-8, 6.5 e-10, 3.5 e-12

La Figure 5 représente la valeur de  $\tilde{\gamma}_k - \gamma^*$  produite par l'APPQ (avec une échelle logarithmique) en fonction du nombre d'itérations  $k$ .

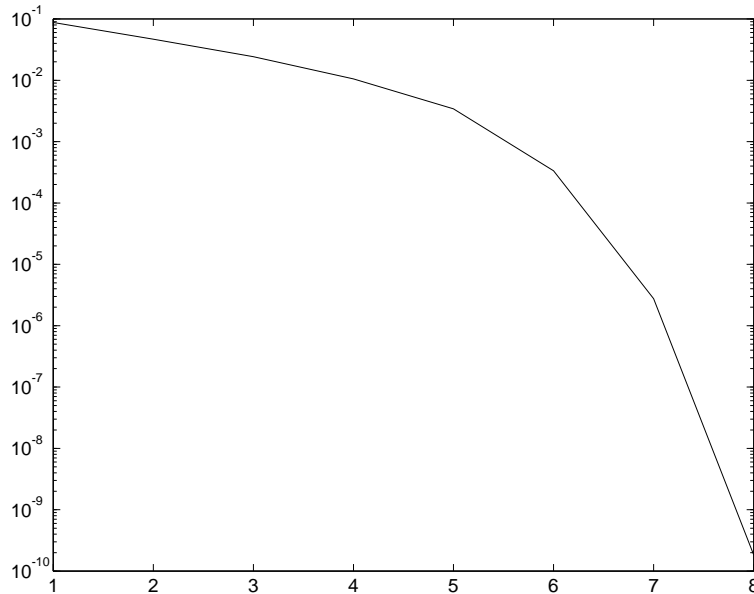


Figure 4:  $\tilde{\gamma}_k - \gamma^*$  en fonction de  $k$ . (Test 3)

On observe que  $\tilde{\gamma}_k$  converge “par paliers”, chacun des paliers correspondant à une valeur propre de  $Q$ .

En fait la convergence est “en moyenne linéaire” sur les 30 premières itérations et elle ne devient véritablement quadratique qu’après le dernier palier. De ce fait, le nombre d’itérations nécessaires pour obtenir une bonne approximation est important malgré une bonne convergence théorique de l’algorithme.

## 6 Conclusion

Les résultats généraux sur les opérateurs maximaux monotones de Rockafellar [8] et de Luque [5] ont été précisés dans le cadre de la minimisation d’une fonction quadratique. Ce cas particulier nous a permis de donner le taux de convergence et de caractériser la limite de la suite générée par l’APP. Une connexion entre l’algorithme

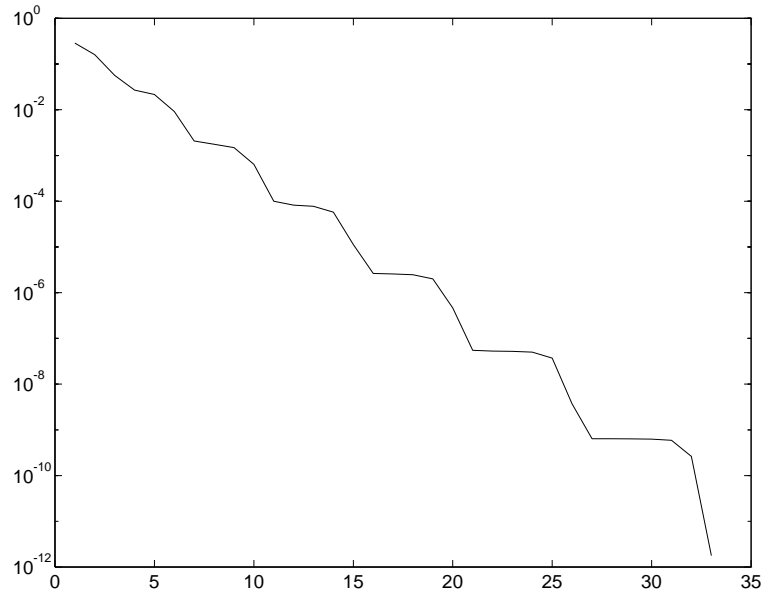


Figure 5:  $\tilde{\gamma}_k - \gamma^*$  en fonction de  $k$ . (Test 4)

proximal et les méthodes classiques d'analyse spectrale a été établie. Enfin, un algorithme proximal a convergence quadratique a été obtenu.

## References

- [1] A. Auslender. Numerical methods for nondifferentiable convex optimization. *Mathematical Programming Study*, 30, pp 102-126, 1987.
- [2] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson. 1988.
- [3] R. Correa, C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62, pp 261-275, 1993.

- [4] P. Lascaux, R. Théodor. Analyse numérique matricielle appliquée à l'art de l'ingénieur. Masson, 1987.
- [5] F.J. Luque. Asymptotic Convergence Analysis of the Proximal Point Algorithm. SIAM Journal on Control and Optimization, vol. 22, No 2, pp 277-293, 1984.
- [6] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. Revue Française d'Informatique et de Recherche Opérationnelle, pp 154-159, 1970.
- [7] N.B. Parlett. The symmetric eigenvalue problem. Prentice Hall. 1980.
- [8] R.T. Rockafellar. Monotone operators and the proximal point algorithm. SIAM Journal on Control and Optimization, 14, pp 877-898, 1976
- [9] R.T. Rockafellar. Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. Math. of Operational Research, pp 97-116, 1976.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)

ISSN 0249-6399