

Optimisation numérique
Méthodes du gradient conjugué linéaire
(Chapitre 5.1)

Daniel KAUTH

Université de Fribourg

le 5 novembre 2009

Table des matières

1	La méthode du gradient conjugué linéaire	1
1.1	Introduction	1
1.2	Quel est le problème à résoudre?	1
1.3	Méthode des directions conjuguées	2
1.4	Propriétés de base de la méthode du gradient conjugué	4
1.5	Une forme plus pratique de la méthode du gradient conjugué	6
1.6	La vitesse de convergence	7
1.7	Préconditionnement	9
1.8	Préconditionneurs pratiques	10

Chapitre 1

La méthode du gradient conjugué linéaire

1.1 Introduction

La méthode présentée dans ce séminaire est celle du gradient conjugué. Il ne s'agit non seulement d'une des techniques les plus utiles pour résoudre de grands systèmes linéaires, mais elle peut même être adaptée de telle manière à ce qu'elle résolve des problèmes d'optimisation non-linéaires. Ces deux variantes, reposant sur la même idée de base, sont respectivement appelées méthodes du gradient conjugué linéaire et non-linéaire.

La méthode analysée ci-dessous est celle trouvée dans les années 50 par Hestens et Stiefel, à savoir la méthode du gradient conjugué linéaire. Cette dernière se base sur la recherche de directions successives permettant d'atteindre la solution exacte d'un système linéaire de matrice symétrique et définie positive et représente une alternative à l'algorithme d'élimination de Gauss. Elle est même souvent préférée à cette dernière lorsque les systèmes d'équations sont grands. Dans la suite, nous allons rencontrer l'effet d'amélioration provoqué par le préconditionnement du système linéaire ainsi qu'un grand nombre d'autres propriétés intéressantes de cette méthode.

1.2 Quel est le problème à résoudre ?

La méthode du gradient conjugué linéaire est une méthode qui résout deux problèmes équivalents possédant la même solution unique. Ces problèmes sont le système d'équations linéaires

$$Ax = b$$

et le problème de minimisation suivant :

$$\Phi(x) = \frac{1}{2}x^T Ax - b^T x.$$

Dans ces deux problèmes, A est une matrice $n \times n$, symétrique et définie positive. Pour la bonne compréhension de la suite de ce séminaire, notons encore que le gradient de Φ est égal au résidu du système linéaire :

$$\nabla\Phi(x) = Ax - b := r(x).$$

1.3 Méthode des directions conjuguées

Une des multiples propriétés intéressantes de la méthode du gradient conjugué est la création économique d'une suite de vecteurs $(p_k)_{k \in \mathbb{N}}$ de n vecteurs A -conjugués, avec A une matrice symétrique définie positive. L'ensemble de ces vecteurs non-nuls de $\mathbb{R}^n \setminus \{p_0, p_2, \dots, p_{n-1}\}$ est dit A -conjugué si

$$p_i^T A p_j = 0, \quad \text{pour tout } i \neq j$$

et forme une base de \mathbb{R}^n . Grâce à l' A -conjugaison des vecteurs de direction $\{p_0, p_2, \dots, p_{n-1}\}$, il est possible de minimiser $\phi(\cdot)$ en seulement n pas, en minimisant $\phi(\cdot)$ successivement selon les n directions. Soient $x_0 \in \mathbb{R}^n$ un point de départ quelconque et $x^* \in \mathbb{R}^n$ la solution exacte d'un problème comme présenté au point (1.1). Définissons la suite

$$x_{k+1} = x_k + \alpha_k p_k \tag{1.3.1}$$

où α_k est le scalaire minimisant $\phi(\cdot)$ selon la direction $x_k + \alpha p_k$ et qui est défini par

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} \tag{1.3.2}$$

Theorème 1. $\forall x_0 \in \mathbb{R}^n$, la suite $\{x_k\}$, définie par l'algorithme des directions conjuguées (défini par (1.3.1) et (1.3.2)), converge en au maximum n pas vers x^* .

Démonstration.

Sachant que p_0, \dots, p_{n-1} engendrent \mathbb{R}^n , définissons $\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}$ t.q.

$$x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \dots + \sigma_{n-1} p_{n-1}.$$

Selon l'algorithme (1.3.1), nous avons que $x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \dots + \alpha_{k-1} p_{k-1}$.

En multipliant cette expression par $p_k^T A$, on voit que $p_k^T A(x_k - x_0) = 0$ et donc que

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - A x_k) = -p_k^T r_k.$$

Ainsi, finalement $\sigma_k = \alpha_k$, ce qui démontre le théorème. □

Il est intéressant de remarquer que si notre matrice A est diagonale, alors les contours de la fonction $\phi(\cdot)$ forment une ellipse dont les axes sont alignés aux directions coordonnées. Dans ce cas, une manière facile et élégante à minimiser $\phi(\cdot)$ est de trouver la solution en minimisant $\phi(\cdot)$ par des minimiseurs, d'une seule dimension chacun, selon les directions coordonnées. Bien sûr, on ne peut pas toujours avoir de la chance : souvent la matrice A n'est pas diagonale. Dans ce cas, les contours de $\phi(\cdot)$ restent une ellipse. Mais une ellipse, dont les axes ne sont en général plus alignés aux directions coordonnées e_i . Ainsi, la solution de $Ax = b$ ne peut plus être trouvée en n itérations et parfois même pas en un nombre fini d'itérations. Pour profiter quand-même du joli comportement décrit ci-dessus, nous allons transformer la matrice A de telle manière à ce qu'elle soit diagonale. Cette transformation est réalisée par introduction de la variable \hat{x} qui est définie par

$$\hat{x} = S^{-1}x$$

avec S la matrice qui a comme colonnes les vecteurs A -conjugués p_0, \dots, p_{n-1} c.à.d. $S := \begin{pmatrix} p_0 & p_1 & \dots & p_{n-1} \end{pmatrix}$.

Nous définissons donc $\hat{\phi}(\hat{x}) := \phi(S\hat{x}) = \frac{1}{2}\hat{x}^T(S^TAS)\hat{x} - (S^Tb)^T\hat{x}$. Ici, on peut facilement voir qu'à cause des vecteurs colonnes A -conjugués, la matrice S^TAS est effectivement diagonale. A partir de ce point, nous utilisons la même procédure qu'avant : nous cherchons une solution en minimisant $\hat{\phi}(\cdot)$ par des minimiseurs, d'une seule dimension chacun, selon les directions coordonnées e_i . Mais cette fois-ci on minimise \hat{x} , au lieu de x . Chaque direction coordonnée dans l' \hat{x} -espace correspond à une direction p_i dans l' x -espace. Nous pouvons donc constater, grâce au Théorème 1, que même dans ce deuxième cas, on aboutit à une solution après au plus n pas de l'algorithme.

Remarquons encore un cas particulier intéressant : lorsque notre matrice A est diagonale et que la matrice de Hess est diagonale aussi, alors les minimiseurs à une dimension, vus lors de la stratégie vue pour une matrice diagonale ci-dessus, déterminent chacun une composante de la solution. Par conséquent, après k minimiseurs à une dimension, on a minimisé ϕ sur un sous-espace de dimension k , par exemple $span\{e_1, \dots, e_k\}$ avec e_i un élément de la base canonique. Avant de passer au Théorème 2, notons encore que

$$r_{k+1} = r_k + \alpha_k Ap_k.$$

Théorème 2. Soit $x_0 \in \mathbb{R}^n$ un point de départ quelconque et soit la suite $\{x_k\}$ générée par l'algorithme des directions conjuguées.

Alors

$$r_k^T p_i = 0 \quad \text{pour } i = 0, \dots, k-1 \tag{1.3.3}$$

et x_k est le minimiseur de $\phi(x)$ sur l'ensemble

$$\{x \mid x = x_0 + span\{p_0, \dots, p_{k-1}\}\} \tag{1.3.4}$$

Démonstration.

Nous montrons d'abord qu'un point \tilde{x} minimise ϕ sur l'ensemble (1.3.4) si et seulement si $r(\tilde{x})^T p_i = 0$, pour tout $i = 0, 1, \dots, k-1$. Définissons $h(\sigma) = \phi(x_0 + \sigma_0 p_0 + \dots + \sigma_{k-1} p_{k-1})$, où $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{k-1})^T$. Comme $h(\sigma)$ est une fonction quadratique strictement convexe, elle a un unique minimiseur, qui satisfait

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = 0 \quad i = 0, 1, \dots, k-1. \tag{1.3.5}$$

Par la règle de la chaîne, ceci implique que

$$\nabla \phi(x_0 + \sigma_0^* p_0 + \dots + \sigma_{k-1}^* p_{k-1})^T p_i = 0 = r(x_k)^T p_i \quad i = 0, 1, \dots, k-1.$$

Ce qui nous donne le résultat.

Ensuite, nous montrons par récurrence que x_k satisfait (1.3.3).

Ancrage : pour $k = 1$, on a $r_1^T p_0 = 0$, par le choix de α_0 dans (1.3.2).

Hypothèse de récurrence : $r_{k-1}^T p_i = 0$ pour $i = 0, 1, \dots, k-2$.

Par

$$r_k = r_{k-1} + \alpha_{k-1} Ap_{k-1}$$

nous obtenons

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \alpha_{k-1} p_{k-1}^T Ap_{k-1} = 0$$

Pour les autres vecteurs p_i , $i = 0, 1, \dots, k-2$, nous avons

$$p_i^T r_k = p_i^T r_{k-1} + \alpha_{k-1} p_i^T Ap_{k-1} = 0$$

par l'hypothèse de récurrence et la conjugaison de p_i . Nous concluons que $r_k^T p_i = 0$, pour $i = 0, 1, \dots, k-1$. \square

Le Théorème 2 nous dit donc que le résidu r_k est orthogonal à toutes les directions de recherche précédentes p_0, p_1, \dots, p_{k-1} .

D'ailleurs, il y a plusieurs méthodes pour choisir les directions de recherche. On pourrait par exemple utiliser les valeurs propres de la matrice A ou encore modifier l'orthogonalisation de Gram-Schmidt à ce que cette procédure produit un ensemble de vecteurs à directions conjugués.

1.4 Propriétés de base de la méthode du gradient conjugué

La méthode du gradient conjugué est une méthode de directions conjuguées qui possède la propriété particulière de n'utiliser que le vecteur p_{k-1} et le résidu r_k pour générer le vecteur p_k de l'ensemble des vecteurs A -conjugués. On a donc à faire avec un algorithme très bon marché en termes d'utilisation de la mémoire.

Dans la méthode du gradient conjugué, les directions p_k sont choisies telles qu'elles sont des combinaisons linéaires entre la direction de la plus grande pente $-\nabla\phi(x_k) = -r_k$ et la direction de p_{k-1} :

$$p_k = -r_k + \beta_k p_{k-1}$$

où β_k est choisi tel que l' A -conjugaison entre p_k et p_{k-1} soit vérifiée. Après multiplication par $p_{k-1}^T A$, on obtient

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

On choisit la direction de la plus grande pente au point initial x_0 comme la première direction de recherche, p_0 . Voici une première version d'un algorithme pour la méthode du gradient conjugué :

Algorithme 3. *Gradient Conjugué- première version*

x_0 soit fixé ;

Posons $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k};$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow Ax_{k+1} - b;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end(while)

Passons maintenant au Théorème 4, qui nous dit que les directions p_0, \dots, p_{n-1} sont effectivement A -conjuguées. D'autant plus, il nous dit que les résidus r_i sont deux à deux orthogonaux et que chaque p_k et chaque résidu r_k sont contenus dans le sous-espace de Krylov de degré k pour r_0 , défini par

$$K(r_0; k) := \text{span}\{r_0, Ar_0, \dots, A^k r_0\}.$$

Théorème 4. *Supposons que le k -ième itéré, généré par la méthode du gradient conjugué, n'est pas le point x^* . Alors les quatre propriétés suivantes sont vraies :*

$$\begin{aligned} r_k^T r_i &= 0, & \text{pour } i=0, \dots, k-1 \\ \text{span}\{r_0, r_1, \dots, r_k\} &= \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \\ \text{span}\{p_0, p_1, \dots, p_k\} &= \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \\ p_k^T A p_i &= 0, & \text{pour } i = 0, 1, \dots, k-1. \end{aligned}$$

Voilà pourquoi la séquence $\{x_k\}$ converge vers x^ en au plus n pas.*

Démonstration. Nous allons démontrer les 3 dernières propriétés par récurrence. Ce qui concerne l'ancrage, les propriétés 2 et 3 sont trivialement vraies pour $k = 0$ et par construction, la propriété 4 est vraie pour $k = 1$. Supposons que ces propriétés soient vraies pour k et montrons qu'elles restent vraies pour $k + 1$. Par ces hypothèses, on a que

$$r_k \in \text{span}\{r_0, Ar_0, \dots, A^k r_0\} \quad \text{et} \quad p_k \in \text{span}\{r_0, Ar_0, \dots, A^k r_0\}$$

En multipliant cette deuxième expression par A et en se rappelant que $r_{k+1} = r_k + \alpha_k A p_k$, on voit facilement que

$$r_{k+1} \in \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}.$$

Grâce à cette constatation et au moyen de l'hypothèse de récurrence de la deuxième propriété, nous avons que

$$\text{span}\{r_0, r_1, \dots, r_{k+1}\} \subset \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}.$$

Par l'hypothèse de récurrence pour la troisième propriété, on peut déduire que

$$A^{k+1} r_0 = A(A^k r_0) \in \text{span}\{A p_0, A p_1, \dots, A p_k\}$$

Comme $r_{k+1} = r_k + \alpha_k A p_k$, il en suit que $A p_i = \frac{(r_{i+1} - r_i)}{\alpha_i}$ et donc

$$A^{k+1} r_0 \in \text{span}\{r_0, r_1, \dots, r_{k+1}\}$$

Ensemble avec l'hypothèse de récurrence de la deuxième propriété, on trouve que

$$\text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\} \subset \text{span}\{r_0, r_1, \dots, r_k, r_{k+1}\}$$

et donc la deuxième propriété est démontrée. Montrons encore que la troisième propriété reste vraie pour $k + 1$:

$$\text{span}\{p_0, p_1, \dots, p_k, p_{k+1}\} = \text{span}\{p_0, p_1, \dots, p_k, r_{k+1}\}, \quad \text{par l'avant-dernière ligne de l'algorithme}$$

$$\begin{aligned}
 &= \text{span}\{r_0, Ar_0, \dots, A^k r_0, r_{k+1}\}, && \text{par l'hypothèse de récurrence de la troisième propriété} \\
 &= \text{span}\{r_0, r_1, \dots, r_k, r_{k+1}\}, && \text{par la deuxième propriété qu'on vient de démontrer} \\
 &= \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}, && \text{par la deuxième propriété pour } k+1
 \end{aligned}$$

Passons à la quatrième propriété à démontrer.

Nous multiplions l'avant dernière ligne de l'algorithme avec Ap_i , $i = 0, 1, \dots, k$ pour obtenir

$$p_{k+1}^T Ap_i = -r_{k+1}^T Ap_i + \beta_{k+1} p_k^T Ap_i \quad (1.4.1)$$

Par la quatrième ligne de l'algorithme, nous avons que si $i = k$, alors $p_{k+1}^T Ap_i = 0$. L'hypothèse de récurrence de la quatrième propriété à démontrer nous dit que p_0, \dots, p_k sont A -conjugués et elle nous permet ainsi d'utiliser le Théorème 2. Il en suit que

$$r_{k+1}^T p_i = 0, \quad \text{pour } i = 0, 1, \dots, k$$

Ensuite, après plusieurs applications de la troisième propriété, on trouve pour $k = 0, \dots, k-1$ que

$$Ap_i \in A \text{span}\{r_0, Ar_0, \dots, A^i r_0\} = \text{span}\{Ar_0, A^2 r_0, \dots, A^{i+1} r_0\} \subset \text{span}\{p_0, \dots, p_{i+1}\}$$

Des deux dernières expressions trouvées, nous concluons que $r_{k+1}^T Ap_i = 0$, pour $i = 0, 1, \dots, k-1$. Le terme de droite de (1.4.1) disparaît donc pour $i = 0, \dots, k-1$. Finalement, par le Théorème 2 et par réarrangement de l'avant dernière ligne de l'algorithme, nous trouvons que

$$p_i = -r_i + \beta_i p_{i-1},$$

tel que $r_i \in \text{span}\{p_i, p_{i-1}\}$ pour tout $i = 1, \dots, k-1$. On conclut que la première affirmation du

Théorème 4 est vérifiée. □

A ce point, il est intéressant de remarquer que le nom 'méthode du gradient conjugué' est mal choisi, car les gradients r_i ne sont qu'orthogonaux alors que ce sont les directions de recherche p_k qui sont A -conjugués. Un meilleur nom pour cette méthode serait donc 'méthode des directions de recherche conjuguées'.

1.5 Une forme plus pratique de la méthode du gradient conjugué

Dans le but de trouver un algorithme plus économique, nous utilisons les résultats des Théorèmes 2 et 4, afin d'arriver à une nouvelle formule pour α_k :

$$\alpha_k = \frac{r_k^T r_k}{p_k^T Ap_k}.$$

Grâce à ceci, il nous est possible d'écrire

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

De ce qui précède, on trouve un deuxième algorithme pour la méthode du gradient conjugué :

Algorithme 5. *Gradient Conjugué* x_0 soit fixé ;Posons $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$;**while** $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k};$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end(while)

Une propriété qui rend la méthode du gradient conjugué particulièrement intéressante est que dans aucune étape de cet algorithme, nous avons besoin de connaître les vecteurs x, r et p pour plus que les deux dernières itérations, les autres ne sont pas stockés.

1.6 La vitesse de convergence

Grâce à la troisième propriété du Théorème 4 et au deuxième pas de l'Algorithme 5, nous avons que

$$x_{k+1} = x_0 + \alpha_0 p_0 + \dots + \alpha_k p_k = x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \dots + \gamma_k A^k r_0.$$

Définissons maintenant P_k^* comme étant un polynôme de degré k avec les coefficients $\gamma_0, \gamma_1, \dots, \gamma_k$. Remplaçons notre polynôme dans l'expression ci-dessus :

$$x_{k+1} = x_0 + P_k^*(A)r_0$$

Nous allons voir maintenant que parmi toutes les méthodes possibles dont les k premiers pas sont restreints sur le sous-espace de Krylov $K(r_0; k)$, l'Algorithme 5 donne le meilleur résultat en termes de minimisation de la distance vers la solution, si cette distance est mesurée par $\| \cdot \|_A$, défini comme

$$\| z \|_A^2 = z^T A z$$

D'après cette définition, on constate que

$$\frac{1}{2} \| x - x^* \|_A^2 = \frac{1}{2} (x - x^*)^T A (x - x^*) = \phi(x) - \phi(x^*).$$

Par conséquent le Théorème 2 nous dit que x_{k+1} minimise $\| x - x^* \|_A^2$ sur l'ensemble $x_0 + \text{span}\{p_0, \dots, p_k\}$. Comme $x_{k+1} = x_0 + P_k^*(A)r_0$, il en suit que P_k^* est la solution de

$$\min_{P_k} \| x_0 + P_k(A)r_0 - x^* \|_A$$

De plus, nous avons que

$$x_{k+1} - x^* = x_0 + P_k^*(A)r_0 - x^* = [I + P_k^*(A)A](x_0 - x^*).$$

Soient $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ les valeurs propres de A et v_1, \dots, v_n les vecteurs propres orthonormés correspondants. Nous savons bien que ces vecteurs propres engendrent \mathbb{R}^n , donc

$$x_0 - x^* = \sum_{i=1}^n \psi_i v_i$$

et il est aussi facile à voir que les vecteurs propres de A sont aussi des vecteurs propres de $P_k(A)$ pour tout polynôme. Nous pouvons donc écrire que $P_k(A)v_i = P_k(\lambda_i)v_i, i = 1, 2, \dots, n$ et donc

$$x_{k+1} - x^* = \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \psi_i v_i$$

La norme définie ci-dessus nous permet de dire que

$$\begin{aligned} \|x_{k+1} - x^*\|_A^2 &= \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \psi_i^2 \\ &\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left(\sum_{j=1}^n \lambda_j \psi_j^2 \right) = \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2, \end{aligned}$$

sachant que $\|x_0 - x^*\|_A^2 = \sum_{j=1}^n \lambda_j \psi_j^2$. Cette dernière expression nous permet de quantifier le taux de convergence de la méthode du gradient conjugué.

Theorème 6. *Si A a seulement r (avec $r < n$) valeurs propres distinctes, alors l'algorithme du gradient conjugué arrive à la solution en au plus r itérations.*

Démonstration. Soient $\tau_1 < \dots < \tau_r$ les valeurs distinctes des valeurs propres $\lambda_1, \dots, \lambda_n$. Définissons maintenant $Q_r(\lambda)$ par

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \dots \tau_r} (\lambda - \tau_1) \dots (\lambda - \tau_r).$$

Il est facile à voir que $Q_r(\lambda) - 1$ est un polynôme de degré r avec une racine $\lambda = 0$. Définissons ensuite \bar{P}_{r-1} , un polynôme de degré $r - 1$ par

$$\bar{P}_{r-1}(\lambda) = (Q_r(\lambda) - 1)/\lambda$$

En posant $k = r - 1$ dans $\min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2$, nous obtenons

$$0 \leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 = \max_{1 \leq i \leq n} Q_r(\lambda_i) = 0$$

Donc, pour $k = r - 1$ on trouve que $\|x_r - x^*\|_A = 0$ et alors $x_r = x^*$. \square

Théorème 7. Si A a des valeurs propres $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, nous avons que

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x_0 - x^*\|_A^2$$

Nous n'allons pas nous occuper de cette preuve. On donne juste une idée au lecteur, comment ce résultat a été trouvé. On choisit un polynôme \bar{P}_k de degré k tel que le polynôme $Q_{k+1}(\lambda) = 1 + \lambda \bar{P}_k(\lambda)$ a comme racines les plus grandes valeurs propres $\lambda_n, \lambda_{n-1}, \dots, \lambda_{n-k+1}$ et le point milieu entre λ_1 et λ_{n-k} . On peut montrer que la valeur maximale atteinte par Q_{k+1} sur $\lambda_1, \dots, \lambda_{n-k+1}$ est $(\lambda_{n-k} - \lambda_1)/(\lambda_{n-k} + \lambda_1)$.

Le Théorème 7 nous permet de prédire le comportement de la méthode du gradient conjugué. Supposons par exemple que les valeurs propres de A consistent en m grandes valeurs propres et les $n - m$ valeurs propres restantes sont situées autour de 1. Si nous définissons $\epsilon = \lambda_{n-m} - \lambda_1$, le Théorème 7 nous dit qu'après $m + 1$ pas de la méthode du gradient conjugué, nous avons que

$$\|x_{m+1} - x^*\|_A \approx \epsilon \|x_0 - x^*\|_A.$$

Pour une petite valeur de ϵ après $m + 1$ pas, la méthode va nous donner une bonne approximation de la solution. Mais attention, le Théorème 7 nous donne une borne supérieure. Il se peut que la méthode nous donne déjà des bons résultats après les premières itérations. Il est généralement vrai que si les valeurs propres apparaissent en r groupes, alors la méthode du gradient conjugué résout approximativement le problème après r pas.

Une autre expression de convergence pour la méthode du gradient conjugué est celle basée sur le nombre de condition spectral. Celui-ci est défini par

$$K = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\lambda_n}{\lambda_1}.$$

On arrive ensuite à montrer que

$$\|x_k - x^*\|_A \leq \left(\frac{\sqrt{K} - 1}{\sqrt{K} + 1}\right)^{2k} \|x_0 - x^*\|_A.$$

Cette expression est plus approximative que celle du Théorème 7 et donne souvent d'importantes surestimations de l'erreur. Mais en réalité, souvent, on n'a pas beaucoup d'informations sur A et ses valeurs propres. Cette dernière méthode ne demande que les valeurs propres extrêmes ou des approximations de celles-ci.

1.7 Préconditionnement

Pour accélérer la méthode du gradient conjugué, on peut transformer le système linéaire de telle façon à ce que les valeurs propres de la matrice A soient mieux distribués. Le processus présenté par la suite est appelé le preconditionnement. Il s'agit de changer la variable x en une variable \hat{x} au moyen d'une matrice C non-singulière : $\hat{x} = Cx$. Par conséquent, ϕ est transformée en

$$\hat{\phi}(\hat{x}) = \frac{1}{2} \hat{x}^T (C^{-T} A C^{-1}) \hat{x} - (C^{-T} b)^T \hat{x} \quad (1.7.1)$$

En utilisant l'Algorithme 5 pour minimiser $\hat{\phi}(\hat{x})$ ou bien pour résoudre le système linéaire

$$(C^{-T}AC^{-1})\hat{x} = C^{-T}b,$$

le taux de convergence dépend des valeurs propres de la matrice $C^{-T}AC^{-1}$ et non de celles de la matrice A . On pourrait donc par exemple choisir C de telle manière que le nombre de condition de $C^{-T}AC^{-1}$ soit plus petit que le nombre de condition de A . Une autre idée serait de choisir C tel que les valeurs propres de $C^{-T}AC^{-1}$ sont regroupées en un petit nombre r de groupes. Ainsi on trouve une bonne approximation après seulement r itérations. L'Algorithme 8 ci-dessous nous n'est rien d'autre qu'une application de l'Algorithme 5 au problème (1.7.1) par rapport à \hat{x} . Ensuite, il transforme toutes les équations pour les exprimer par rapport à x . On remarque que l'Algorithme 8 n'utilise pas explicitement C , mais plutôt la matrice $M = C^T C$, qui est symétrique et définie positive par construction.

Algorithme 8. *Gradient Conjugué Préconditionné*

x_0 soit fixé et M le préconditionneur de A_i ;

Posons $r_0 \leftarrow Ax_0 - b$;

Résoudre $My_0 = r_0$ par rapport à y_0 ; Posons $p_0 \leftarrow -r_0, k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k};$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

$$M y_{k+1} \leftarrow r_{k+1}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T r_k};$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end(while)

Si dans l'Algorithme 8, nous remplaçons M par I , on retombe sur l'Algorithme 5. Les propriétés de l'Algorithme 5 se généralisent de façon intéressante. Par exemple la première affirmation du Théorème 4, à savoir $r_k^T r_i = 0$, pour $i = 0, \dots, k-1$ devient

$$r_i^T M^{-1} r_j = 0, \quad \text{pour tout } i \neq j .$$

Du point de vue de l'ordinateur, la grande différence entre l'Algorithme 8 et l'Algorithme 5 est qu'il faut encore résoudre des systèmes de la forme $My = r$.

1.8 Préconditionneurs pratiques

Comme si souvent en numérique, il n'existe pas une meilleure stratégie de préconditionnement valable pour tout type de matrices. Des préconditionneurs généraux ont bien été proposés, mais leur efficacité varie fortement d'un problème à l'autre. Les stratégies les plus importantes de ce genre sont la SSOR (symmetric successive overrelaxation), la méthode de Cholesky incomplète et les préconditionneurs par bande.