Résolution itérative de systèmes linéaires

J. Erhel

Janvier 2014

Dans tout ce chapitre, on cherche à résoudre le système

$$Ax = b, \tag{1}$$

où $A \in \mathbb{R}^{n \times n}$ est une matrice inversible et $b \in \mathbb{R}^n$ le second membre. On note x^* la solution.

Dans ces méthodes itératives, la matrice n'est pas transformée, seule l'opération produit matrice-vecteur $y = Ax$ est requise. Il est donc possible d'utiliser des versions dites "matrix-free" où la matrice n'est pas stockée. Les méthodes itératives nécessitent en général moins d'espace mémoire que les méthodes directes.

Dans tout ce qui suit, on note x_k l'approximation de la solution à l'itération k, le résidu est défini par $r_k = b - Ax_k$ et $e_k = x^* - x_k$ est l'erreur, de sorte que $r_k = Ae_k$.

1 Méthodes itératives linéaires

Les méthodes itératives basiques sont les méthodes appelées ici linéaires. Elles ne sont plus beaucoup utilisées en soi car leur convergence n'est pas toujours assurée et est en général lente, surtout pour les systèmes de grande taille. Elles servent par contre à accélérer la convergence d'une classe importante de méthodes itératives, appelées méthodes polynomiales. Pour une description détaillée de ces méthodes, on pourra consulter [2, 3, 14, 20].

Les méthodes linéaires sont basées sur une décomposition

$$A = M - N, \tag{2}$$

où M est une matrice inversible. L'itération est un schéma de point fixe, défini, à partir de x_0 donné, par

$$Mx_{k+1} = Nx_k + b. \tag{3}$$

On obtient donc $Mx_{k+1} = (M - A)x_k + b = Mx_k + r_k$ soit

$$x_{k+1} = x_k + M^{-1}r_k.$$

On a aussi $Mx_{k+1} = Nx_k + (M - N)x^*$ d'où $M(x^* - x_{k+1}) = N(x^* - x_k)$ donc

$$e_{k+1} = (M^{-1}N)e_k.$$

Théorème 1.1 *Soit ρ le rayon spectral (la plus grande valeur propre en valeur absolue) de la matrice $M^{-1}N$, correspondant à la décomposition $A = M - N$ de la matrice inversible A. La méthode (3) est convergente pour tout vecteur initial x_0 si et seulement si $\rho < 1$.*

Les exemples les plus classiques sont les méthodes de Jacobi, Gauss-Seidel, SOR et SSOR.

Soit $A = D - E - F$ la décomposition de A en parties diagonale, triangulaire inférieure et triangulaire supérieure.

La méthode de Jacobi est définie par $M = D$, celle de Gauss-Seidel par $M = D - E$. La méthode SOR (Successive Over Relaxation) résout $\omega Ax = \omega b$ avec la décomposition $M = D - \omega E$, où ω est un paramètre choisi pour accélérer la convergence. On a les itérations

$$\begin{aligned} Jacobi &: Dx_{k+1} = (E+F)x_k + b \\ Gauss-Seidel &: (D-E)x_{k+1} = Fx_k + b \\ SOR &: (D-\omega E)x_{k+1} = (\omega F + (1-\omega)D)x_k + \omega b \end{aligned}$$

La méthode SSOR (Symmetric SOR) effectue une itération SOR suivie d'une itération SOR en sens inverse. On obtient

$$M = \frac{1}{\omega(2-\omega)}(D - \omega E)D^{-1}(D - \omega F)$$

Il est possible de prouver la convergence pour certaines classes de matrices. Pour la démonstration de ces théorèmes et pour d'autres résultats, voir par exemple [10, 26].

Définition 1.1 *Une matrice $A = (a_{ij})$ d'ordre n est à diagonale strictement dominante si*

$$|a_{jj}| > \sum_{i=1, i\neq j}^{n} |a_{ij}|, \ j = 1, \ldots, n$$

Théorème 1.2 *Si A est à diagonale strictement dominante, les méthodes de Jacobi et de Gauss-Seidel convergent pour tout x_0.*

Théorème 1.3 *Soit A une matrice symétrique avec des éléments diagonaux positifs et soit $0 < \omega < 2$. La méthode SOR converge pour tout x_0 si et seulement si A est définie positive.*

2 Méthodes de projection

2.1 Définition d'une méthode polynomiale

La proposition suivante explique pourquoi il est intéressant de définir des approximations polynomiales de la solution d'un système linéaire.

Proposition 2.1 *Il existe un polynôme p de degré au plus $n-1$ tel que $A^{-1} = p(A)$.*

Preuve. D'après le théorème de Cayley-Hamilton, le polynôme caractéristique q de A, defini par $q(x) = det(A - xI)$, est de degré n et vérifie $q(A) = 0,\ q(0) = det(A)$, où $det(A)$ est le déterminant de A.

Soit $q(X) = det(A) + Xq_1(X)$ alors $Aq_1(A) = -det(A)I$.

Puisque A est inversible, $det(A) \neq 0$ et $-\frac{1}{det(A)}Aq_1(A) = I$.

Soit p le polynôme défini par $p(X) = -\frac{1}{det(A)}q_1(X)$, alors $A^{-1} = p(A)$. $\diamond$

Soit x_0 une approximation initiale et $r_0 = b - Ax_0$. Alors la solution x^* vérifie $x^* = x_0 + A^{-1}r_0 = x_0 + p(A)r_0$.

L'objectif des méthodes polynomiales est d'approcher ce polynôme $p(X)$. Il existe deux directions pour définir une méthode polynomiale :

- La première direction consiste à définir explicitement les polynômes. C'est le cas de la méthode CHEBY qui s'applique au cas où la matrice est symétrique définie positive [2].
- La deuxième manière de considérer une méthode polynomiale est de manipuler les polynômes indirectement, à travers leur application à la matrice et à un vecteur. Dans ce cas, on ne calcule pas les coefficients du polynôme. La plupart des méthodes polynomiales sont basées sur cette approche.

Définition 2.1 *L'espace de Krylov associé à A et r_0 est défini par*

$$\mathcal{K}_k(A, r_0) = eng\{r_0, Ar_0, \ldots, A^{k-1}r_0\} = \{s_{k-1}(A)r_0, \quad \text{où } s_{k-1} \text{ est un polynôme de degré au plus } k-1\}.$$

Définition 2.2 *Une méthode itérative est polynomiale si et seulement si elle vérifie la condition de sous-espace*

$$x_k = x_0 + s_{k-1}(A)r_0, \tag{4}$$

où s_{k-1} est un polynôme de degré au plus $k-1$.

De manière équivalente, la condition peut s'écrire

$$x_k \in x_0 + \mathcal{K}_k(A, r_0),$$

ou

$$x_{k+1} \in x_k + \mathcal{K}_{k+1}(A, r_0). \tag{5}$$

C'est une méthode de sous-espace liée aux espaces de Krylov.

Proposition 2.2 *Dans une méthode polynomiale, le résidu r_k et l'erreur e_k vérifient*

$$r_k = q_k(A)r_0, \quad e_k = q_k(A)e_0, \tag{6}$$

où $q_k(X) = 1 - Xs_{k-1}(X)$ est un polynôme de degré au plus k qui vérifie $q_k(0) = 1$.

Preuve. $r_k = b - A(x_0 + s_{k-1}(A)r_0) = r_0 - As_{k-1}(A)r_0 = q_k(A)r_0$ où $q_k(X) = 1 - Xs_{k-1}(X)$.
$e_k = A^{-1}r_k = A^{-1}q_k(A)r_0 = q_k(A)A^{-1}r_0 = q_k(A)e_0$. ◇

Définition 2.3 *L'ensemble des polynômes résiduels de degré k est défini par*

$$\mathcal{P}_k^0 = \{q \text{ est un polynôme de degré } k \text{ et } q(0) = 1\}.$$

Par la proposition suivante, on montre que les méthodes polynomiales, définies à partir des espaces de Krylov, engendrent des suites finies et calculent la solution exacte. Donc ces méthodes sont en fait des méthodes directes, mais elles sont utilisées en tant que méthodes itératives. Ce paradoxe n'est qu'apparent, car il s'agit de converger vers une approximation précise bien avant d'aboutir à la solution exacte.

Proposition 2.3 *La suite des sous-espaces de Krylov $\mathcal{K}_k(A, r_0)$ est une suite croissante de sous-espaces, et donc stationnaire à partir d'un rang $p \leq n$; pour $k \leq p$, la dimension de $\mathcal{K}_k(A, r_0)$ est égale à k, et le système $\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ est une base de $\mathcal{K}_k(A, r_0)$.*

Le sous-espace $\mathcal{K}_p(A, r_0)$ est un sous-espace invariant de A; la solution de $Ay = r_0$ appartient à cet espace.

Preuve. La première partie de la proposition est évidente puisque tous les sous-espaces sont au plus de dimension n, dimension de A. Soit p le premier rang où la dimension de $\mathcal{K}_{k+1}(A, r_0)$ est égale à k. Alors $\mathcal{K}_{p+1}(A, r_0) = \mathcal{K}_p(A, r_0)$, et la suite est stationnaire à partir de ce rang. On a donc :

$$A\mathcal{K}_p(A, r_0) \subset \mathcal{K}_{p+1}(A, r_0) = \mathcal{K}_p(A, r_0),$$

ce qui pouve l'invariance de l'espace. L'opérateur A définit donc une bijection de l'espace dans lui-même. Le vecteur r_0 appartenant à $\mathcal{K}_p(A, r_0)$, il a un antécédent dans cet espace. ◇

En fait, pour les grands systèmes, on cherche à arrêter les itérations avant d'aboutir à la solution exacte. L'objectif qui guide la construction des méthodes de Krylov est une propriété de minimisation de l'erreur e_k, pour un certain produit scalaire. Cet objectif n'est pas toujours réalisable, c'est pourquoi on définit une condition plus générale d'orthogonalité.

2.2 Méthodes polynomiales de projection

Définition 2.4 *Une méthode de projection de Krylov est une méthode polynomiale définie par une matrice B et deux conditions : la condition de sous-espace (5) et la condition dite de Petrov-Galerkin :*

$$(Be_k)^T u = 0, \ \forall u \in \mathcal{K}_k(A, r_0). \tag{7}$$

Le choix de B dicte la construction de la méthode de projection. Lorsque la matrice B définit un produit scalaire, on obtient la propriété de minimisation souhaitée :

Proposition 2.4 *Si B est une matrice symétrique définie positive alors, pour x_k satisfaisant la condition de sous-espace (5), la condition de Petrov-Galerkin (7) est équivalente à minimiser l'erreur :*

$$\|e_k\|_B = \min_{y \in \mathcal{K}_k(A, r_0)} \|e_0 - y\|_B, \tag{8}$$

où $e_k = x^* - x_k = x^* - (x_0 + y) = e_0 - y$. Dans ce cas x_k est déterminé de manière unique.

Preuve. Soit $x_k = x_0 + y$ avec $y \in \mathcal{K}_k(A, r_0)$. La condition (7) exprime que l'erreur $e_k = e_0 - y$ doit appartenir au B-orthogonal de l'espace $\mathcal{K}_k(A, r_0)$. Or $e_k \in e_0 + \mathcal{K}_k(A, r_0)$. Il s'ensuit que e_k est la projection B-orthogonale de e_0 sur $(\mathcal{K}_k(A, r_0))^{\perp_B}$. Le vecteur y solution du problème aux moindres carrés (8) est bien la projection B-orthogonale de e_0 sur $\mathcal{K}_k(A, r_0)$. ◇

2.3 Préconditionnement d'une méthode polynomiale

La vitesse de convergence des méthodes itératives polynomiales dépend de certaines propriétés de la matrice A. Une façon d'accélérer la convergence est de transformer le problème (1) en le préconditionnant à l'aide d'une matrice connue. Le nouveau système à résoudre est équivalent mais l'objectif est que la méthode itérative polynomiale converge plus rapidement.

Définition 2.5 *Un préconditionnement est défini par une matrice inversible $C \in \mathbb{R}^{n,n}$. Un préconditionnement à gauche résout le système équivalent*

$$CAx = Cb,$$

tandis qu'un préconditionnement à droite résout le système équivalent

$$AC(C^{-1}x) = b.$$

Dans un système préconditionné à gauche, la matrice est donc CA, de sorte que le produit $v = Au$ est remplacé par la séquence des deux produits $w = Au, v = Cw$.

Si $C = A^{-1}$, alors $CA = I$ de sorte que le système préconditionné est trivial. Le préconditionnement C est choisi pour approcher A^{-1}.

Proposition 2.5 *Une méthode polynomiale préconditionnée à gauche est caractérisée par*

$$x_k \in x_0 + \mathcal{K}_k(CA, Cr_0),$$

et une méthode polynomiale préconditionnée à droite est caractérisée par

$$x_k \in x_0 + C\mathcal{K}_k(AC, r_0).$$

Preuve. Considérons le système $CAx = Cb$. Soit x_0 le choix initial de la méthode polynomiale associée. Alors le résidu s_0 vérifie $s_0 = Cb - CAx_0 = C(b - Ax_0) = Cr_0$ et le polynôme p_k est appliqué à CA.

La preuve est similaire pour le système préconditionné à droite. ◇

Les méthodes linéaires vues précédemment sont en fait des méthodes polynomiales préconditionnées.

Proposition 2.6 *Une méthode linéaire est une méthode polynomiale préconditionnée à droite. Le préconditionnement est $C = M^{-1}$ et le polynôme résiduel est $q(X) = (1 - X)^k$.*

Preuve. $r_{k+1} = r_k - AM^{-1}r_k = (I - AM^{-1})r_k = (I - AM^{-1})^k r_0$. ◇

3 Cas où A est symétrique définie positive

Nous avons vu que la méthode SOR converge si A est symétrique définie positive. Toutefois, la vitesse de convergence est en général très lente. Il est préférable d'utiliser la méthode de Gradient Conjugué.

3.1 Méthode du Gradient Conjugué

La méthode du Gradient Conjugué (GC) est la méthode polynomiale de choix dans le cas où A est symétrique définie positive. De nombreux ouvrages décrivent cette méthode. Nous décrivons la méthode GC dans le cadre des méthodes de projections de Krylov.

Définition 3.1 *La méthode du gradient conjugué (GC) est la méthode de projection de Krylov dans laquelle $B = A$ et A est symétrique définie positive.*

La proposition 2.4 se traduit immédiatement par

Proposition 3.1 *Les itérés x_k du gradient conjugué sont bien définis pour tout $k \geq 0$,à partir des conditions d'espace et de Petrov-Galerkin. Ils vérifient*

$$\|e_k\|_A = \|r_k\|_{A^{-1}} = \min_{y_i n \mathcal{K}_k} \|e_0 - y\|_A.$$

Il existe $p \leq n$ tel que $x_p = x^*$.

Définition 3.2 *D'après la condition d'espace (5), il existe $\alpha_k \in \mathbb{R}$ et $p_k \in \mathcal{K}_{k+1}(A, r_0)$ tels que*

$$x_{k+1} = x_k + \alpha_k p_k. \tag{9}$$

Le vecteur p_k est appelé direction de descente.

Proposition 3.2 *La condition de Galerkin (7) est équivalente à*

$$r_k \perp \mathcal{K}_k(A, r_0). \tag{10}$$

Preuve. Evidente, puisque $Be_k = Ae_k = r_k$. $\diamond$

Théorème 3.1 *Tant que les résidus r_k sont non nuls, la méthode GC vérifie*

$$\begin{array}{l} r_0 = b - Ax_0, \\ x_{k+1} = x_k + \alpha_k p_k, \\ r_{k+1} = r_k - \alpha_k A p_k, \\ eng\{r_0, r_1, \ldots, r_k\} = eng\{p_0, p_1, \ldots, p_k\} = \mathcal{K}_{k+1}(A, r_0), \\ r_k^T r_j = 0, \ j \neq k, \\ p_k^T A p_j = 0, \ j \neq k, \end{array}$$

et les directions de descente peuvent être choisies pour vérifier la récurrence (à une constante près)

$$p_0 = r_0, \ p_{k+1} = r_{k+1} + \beta_k p_k. \tag{11}$$

Preuve. La condition (9) implique $r_{k+1} = r_k - \alpha_k A p_k$. On a $r_k \in \mathcal{K}_{k+1}(A, r_0)$ et

$$\begin{array}{l} eng\{r_0, r_1, \ldots, r_k\} \subseteq \mathcal{K}_{k+1}(A, r_0), \\ eng\{p_0, p_1, \ldots, p_k\} \subseteq \mathcal{K}_{k+1}(A, r_0). \end{array}$$

La condition de Galerkin (10) et le fait que les résidus sont dans l'espace de Krylov impliquent immédiatement l'orthogonalité des résidus.

Si $r_j \neq 0, \ j = 0, 1, \ldots, k$, alors le sous-espace $eng\{r_0, r_1, \ldots, r_k\}$ est de dimension $k+1$ (base orthogonale) et on obtient l'égalité des sous-espaces. Le sous-espace de Krylov $\mathcal{K}_k(A, r_0)$ est donc de dimension k.

La condition de Galerkin et la récurrence $r_{k+1} = r_k - \alpha_k A p_k$ induisent $p_j^T A p_k = 0, \ j \leq k-1$. Puisque A est symétrique, on obtient $p_k^T A p_j = 0, \ j \leq k-1$.

Puisque $p_0 \in eng\{r_0\}$, on peut choisir $p_0 = r_0$.

Puisque $r_{k+1} \in eng\{p_0, p_1, \ldots, p_{k+1}\}$, on peut écrire

$$r_{k+1} = \sum_{i=0}^{k+1} \gamma_i p_i.$$

Pour $j \leq k-1, \ Ap_j \in \mathcal{K}_{j+2} \subseteq \mathcal{K}_{k+1}$ donc $r_{k+1}^T A p_j = 0$. D'autre part, $p_i^T A p_j = 0, i \neq j$ d'où $\gamma_j = 0, j \leq k-1$. On en déduit que

$$r_{k+1} = \gamma_k p_k + \gamma_{k+1} p_{k+1}.$$

Puisque $dim(\mathcal{K}_k) = k$, on a $\gamma_{k+1} \neq 0$ et on choisit $\gamma_{k+1} = 1$. On en déduit la récurrence (11). $\diamond$

Remarque 3.1 *L'hypothèse que A est symétrique permet de montrer la récurrence courte (11) sur les directions de descente, qui relie p_{k+1} à seulement p_k et non aux p_j.*

Il reste à caractériser α_k et β_k pour définir complètement la méthode.

Théorème 3.2 *On considère le système $Ax = b$, où A une matrice symétrique définie positive. La méthode du Gradient Conjugué est définie par l'algorithme 1.*

ALGORITHM 1: CG

* Initialisation ;
choisir x_0 ;
$r_0 = b - Ax_0$;
$p_0 = r_0$;
$\rho_0 = \|r_0\|^2$;
* Iterations ;
for $k = 0, 1, \ldots$ **until** convergence **do**
$q_k = Ap_k$;
$\alpha_k = \frac{rho_k}{p_k^T q_k}$;
$x_{k+1} = x_k + \alpha_k p_k$;
$r_{k+1} = r_k - \alpha_k q_k$;
$\rho_{k+1} = \|r_{k+1}\|_2^2$;
$\beta_k = \frac{\rho_{k+1}}{\rho_k}$;
$p_{k+1} = r_{k+1} + \beta_k p_k$;
end do

Preuve. Les récurrences sont déjà prouvées.

Les résidus sont orthogonaux donc, en particulier, $r_{k+1}^T r_k = 0$, d'où $r_k^T r_k - \alpha_k r_k^T A p_k = 0$. Mais $r_k = p_k - \beta_{k-1} p_{k-1}$ et les directions de descente sont A-conjuguées donc $r_k^T A p_k = p_k^T A p_k$ et $\alpha_k = r_k^T r_k / p_k^T A p_k$.

Les directions de descente sont conjuguées donc, en particulier, $p_{k+1}^T A p_k = 0$. D'autre part,

$$r_{k+1}^T A p_k = \frac{1}{\alpha_k} r_{k+1}^T (r_k - r_{k+1}) = -\frac{1}{\alpha_k} r_{k+1}^T r_{k+1}$$

d'où

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{\alpha_k p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

Réciproquement, si les suites sont définies par les récurrences ci-dessus, il est aisé de montrer par récurrence que la méthode vérifie les conditions d'espace (5) et de Petrov-Galerkin (7). ◇

Proposition 3.3 *Les coefficients β_k et α_k existent et sont uniques tant que $r_k \neq 0$.*

Preuve. Tant que $r_k \neq 0$, le coefficient β_k est bien défini et est unique. Tant que $r_k \neq 0$, on a aussi $p_k \neq 0$. La matrice A est définie positive donc $p_k^T A p_k \neq 0$ et le coefficient α_k existe et est unique. ◇

Remarque 3.2 *L'hypothèse que A est définie positive garantit l'existence et l'unicité de α_k.*

Le nombre d'itérations *niter* est inconnu. Par contre, on peut calculer le nombre d'opérations de chaque itération. Soit N le coût du produit matrice-vecteur. Si la matrice A est pleine, le produit est une opération BLAS2 et $N = 2n^2 + O(n)$. Si la matrice A est creuse avec un stockage compressé par lignes ou par colonnes, $N = 2nz(A) + O(n)$. Les autres opérations sont des opérations vectorielles, de type BLAS1.

Le coût global de chaque itération est $N + 10n$.

3.2 Lien avec la méthode de Lanczos

La méthode de Lanczos symétrique construit itérativement une base orthonormée de l'espace de Krylov ayant un vecteur v_1 comme départ. La base orthonormée V_k vérifie $V_k^T A V_k = T_k$, où T_k est une matrice tridiagonale. L'objet de ce paragraphe est de montrer que la méthode de Gradient Conjugué est une variante de la méthode de Lanczos.

Lemme 3.1 *Soit $v_{j+1} = \frac{r_j}{\|r_j\|}$, $j = 0, 1, \ldots$ et $V_k = (v_1, \ldots, v_k)$. Alors le système V_k est une base orthonormée de l'espace de Krylov $\mathcal{K}_k(A, r_0)$ et*

$$Av_{k+1} = \delta_k v_k + \gamma_{k+1} v_{k+1} + \delta_{k+1} v_{k+2}, \tag{12}$$

où

$$\delta_k = -\frac{\sqrt{\beta_{k-1}}}{\alpha_{k-1}}, \quad \gamma_{k+1} = \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}}.$$

Preuve. les vecteurs v_k sont orthonormés et engendrent l'espace de Krylov.

Les relations de récurrence s'écrivent

$$\begin{aligned}
&r_k = p_k - \beta_{k-1} p_{k-1} \text{ et } Ap_k = \tfrac{1}{\alpha_k}(r_k - r_{k+1}),\\
&\text{d'où}\\
&Ar_k = \tfrac{1}{\alpha_k}(r_k - r_{k+1}) - \tfrac{\beta_{k-1}}{\alpha_{k-1}}(r_{k-1} - r_k),\\
&Ar_k = -\tfrac{\beta_{k-1}}{\alpha_{k-1}} r_{k-1} + (\tfrac{1}{\alpha_k} + \tfrac{\beta_{k-1}}{\alpha_{k-1}}) r_k - \tfrac{1}{\alpha_k} r_{k+1},
\end{aligned}$$

d'où la relation (12). ◇

De ce lemme, nous déduisons l'équivalence entre le Gradient Conjugué et une méthode de Lanczos.

Théorème 3.3 *La méthode GC construit une base orthonormée $V_k = (v_1, \ldots, v_k)$ de l'espace de Krylov $\mathcal{K}_k(A, r_0)$ qui vérifie*

$$AV_k = V_k T_k + \delta_k v_{k+1} u_k^T \tag{13}$$

où $T_k \in \mathbb{R}^{k \times k}$ est une matrice tridiagonale symétrique et où $u_k^T = (0 \ldots 0\ 1) \in \mathbb{R}^k$.

Autrement dit, la méthode GC applique la méthode de Lanczos au vecteur de départ $v_1 = r_0/\|r_0\|$.

La méthode GC calcule $x_k = x_0 + V_k y$ *où* $y \in \mathbb{R}^k$ *est solution du système linéaire*

$$T_k y = \|r_0\|_2 u_1, \tag{14}$$

où $u_1^T = (1\ 0 \dots 0) \in \mathbb{R}^k$.

Preuve. Posons

$$T_k = \begin{pmatrix} \gamma_1 & \delta_1 & & \\ \delta_1 & \gamma_2 & \delta_2 & \\ & . & . & . \\ & & \delta_{k-1} & \gamma_k \end{pmatrix},$$

alors, d'après (12), $AV_k = V_k T_k + \delta_k v_{k+1} u_k^T$.

Cette propriété (13) est caractéristique de la méthode de Lanczos.

Soit $x_k = x_0 + V_k y$ alors $r_k = r_0 - AV_k y$ et $V_k^T r_k = 0$ s'écrit $V_k^T A V_k y = V_k^T r_0$. Or $V_k^T A V_k = T_k$ et $V_k^T r_0 = \|r_0\|_2 u_1$. ◇

Remarque 3.3 *La matrice* T_k *est symétrique définie positive, comme la matrice* A.

Preuve. $y^T T_k y = (V_k y)^T T_k (V_k y)$. ◇

En fait, le Gradient Conjugué effectue par récurrence une factorisation de la matrice T_k.

3.3 Convergence de GC

Nous avons vu la construction de la méthode du Gradient Conjugué. Nous allons maintenant étudier les propriétés de convergence.

Il peut paraître paradoxal d'étudier la vitesse de convergence d'une méthode qui se termine en un nombre fini d'itérations. Néanmoins, en pratique, l'ordre n est très grand et la méthode du Gradient Conjugué fournit une approximation assez précise en beaucoup moins d'itérations. Nous établissons un premier résultat sur la décroissance des normes d'erreur. Il est à noter que c'est la norme A de l'erreur, donc la norme A^{-1} du résidu, qui décroît strictement.

Soit $\sigma(A)$ l'ensemble des valeurs propres de A, avec $0 < \lambda_1 \leq \dots \leq \lambda_n$ de A. Soit

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \lambda_n/\lambda_1$$

le conditionnement spectral de A.

Théorème 3.4 *La méthode GC a une convergence strictement monotone, plus précisément*

$$\|e_{k+1}\|_A \leq (1 - \frac{1}{\kappa(A)})^{1/2} \|e_k\|_A. \tag{15}$$

Preuve.

$r_{k+1} = r_k - \alpha_k A p_k$,
$\|e_{k+1}\|_A^2 = e_{k+1}^T A e_{k+1} = r_{k+1}^T A^{-1} r_{k+1} = r_k^T A^{-1} r_k - 2\alpha_k r_k^T p_k + \alpha_k^2 p_k^T A p_k$
or $p_k^T A p_k = \frac{r_k^T r_k}{\alpha_k}$
et $r_k^T p_k = r_k^T r_k$
d'où $\|e_{k+1}\|_A^2 = \|e_k\|_A^2 - \alpha_k \|r_k\|_2^2$.

Nous allons minorer successivement $\|r_k\|_2^2$ et α_k. Nous avons

$\|e_k\|_A^2 = r_k^T A^{-1} r_k \leq \|A^{-1}\|_2 \|r_k\|_2^2$.
D'autre part,
$p_k^T A p_k = (r_k + \beta_{k-1} p_{k-1})^T A p_k = r_k^T A p_k = r_k^T A r_k + \beta_{k-1} r_k^T A p_{k-1}$
or $r_k^T A p_{k-1} = (p_k - \beta_{k-1} p_{k-1})^T A p_{k-1} = -\beta_{k-1} p_{k-1}^T A p_{k-1} \leq 0$
donc $p_k^T A p_k \leq r_k^T A r_k \leq \|A\|_2 \|r_k\|_2^2$ et $\alpha_k = \frac{\|r_k\|_2^2}{p_k^T A p_k} \geq \frac{1}{\|A\|_2}$.

On en conclut que

$\alpha_k \|r_k\|_2^2 \geq \frac{1}{\|A\|_2 \|A^{-1}\|_2} \|e_k\|_A^2$
donc $\|e_{k+1}\|_A^2 \leq (1 - \frac{1}{\kappa(A)}) \|e_k\|_A^2$.

$\diamond$

Le facteur de décroissance dépend du conditionnement. Si celui-ci est grand, le coefficient devient proche de 1 et la convergence risque d'être lente.

Nous allons maintenant établir une majoration de l'erreur améliorant le résultat ci-dessus. Pour cela, nous allons exploiter la propriété de minimisation de la norme de l'erreur et le caractère polynomial de la méthode.

Théorème 3.5 *Les itérations de GC vérifient*

$$\|e_k\|_A = \min_{q \in \mathcal{P}_k^0} \|q(A) e_0\|_A.$$

Preuve. L'ensemble $\{\|q(A)e_0\|_A,\ q \in \mathcal{P}_k^0\}$ n'est rien d'autre que l'ensemble $\{\|b - Ay\|_{A^{-1}},\ y \in x_0 + \mathcal{K}_k(A, r_0)\}$. La propriété de minimisation est donc équivalente sur l'espace de Krylov et sur l'espace des polynômes résiduels. $\diamond$

Nous pouvons maintenant traduire cette propriété sous la forme dite "min-max", en utilisant la décomposition de A en valeurs propres.

Corollaire 3.1 *Les itérations du GC vérifient*

$$\|e_k\|_A \leq \|e_0\|_A \min_{q \in \mathcal{P}_k^0} \max_{z \in \sigma(A)} |q(z)|, \tag{16}$$

Preuve. Soit $A = U\Sigma U^{-1}$ la décomposition spectrale de A, où Σ est la matrice diagonale $diag(\lambda_1, \ldots, \lambda_n)$, avec $\{\lambda_i,\ i = 1, \ldots, n\}$ les valeurs propres strictement positives de A et U est la matrice orthogonale de colonnes les vecteurs propres de A. On a

$$\begin{array}{l} A^m = U\Sigma^m U^{-1},\ q(A) = Uq(\Sigma)U^{-1}, \\ e_0 = \sum_{i=1}^n \mu_i u_i = U\mu, \\ \|e_0\|_A^2 = e_0^T A e_0 = \sum_{i=1}^n \mu_i^2 \lambda_i, \\ q(A)e_0 = \sum_{i=1}^n q(\lambda_i)\mu_i u_i, \\ \|q(A)e_0\|_A \leq \max_i |q(\lambda_i)| \|e_0\|_A. \end{array}$$

En utilisant le théorème 3.5, on en déduit la relation (16). ◇

La propriété "min-max" permet d'utiliser la théorie de l'approximation sur les polynômes et de calculer une majoration de l'erreur.

Corollaire 3.2 *Les itérations de GC vérifient*

$$\|e_k\|_A \leq 2\|e_0\|_A (\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1})^k. \tag{17}$$

Preuve. Nous utilisons, sans le démontrer, le fait que

$$\min_{q \in \mathcal{P}_k^0} \max_{z \in [\lambda_1, \lambda_n]} |q(z)| = \frac{1}{|C_k(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1})|},$$

où C_k est le polynôme de Chebyshev de première espèce de degré k. Voir par exemple [20]. D'où le résultat du théorème. ◇

Remarque 3.4 *La borne (17) est meilleure que la borne (15). En effet,*

$$\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \leq \frac{\sqrt{\kappa(A) - 1}}{\sqrt{\kappa(A)}} = (1 - 1/\kappa(A))^{1/2}$$

car $\sqrt{\kappa(A)} - 1 \leq \sqrt{\kappa(A) - 1}$.

3.4 Convergence superlinéaire

Nous avons vu que la convergence du Gradient Conjugué est liée aux valeurs propres. D'autre part, les valeurs propres de la matrice T_k sont les valeurs de Ritz et convergent vers les valeurs propres de A (voir méthode de Lanczos). Lorsqu'une valeur extrémale du spectre a convergé, les itérations du Gradient Conjugué se poursuivent comme s'il y avait eu une déflation dans le spectre de A. Tout se passe comme si le conditionnement était plus petit, par exemple λ_{n-1}/λ_1 et la convergence s'accélère. Puis une autre valeur de Ritz converge vers λ_{n-1} et la convergence s'accélère de nouveau. Ce phénomène s'appelle convergence superlinéaire. Il est étudié en détails dans [22, 24].

3.5 Gradient Conjugué Préconditionné

Soit A une matrice symétrique définie positive et C une matrice inversible. Les systèmes préconditionnés définis dans 2.5 ne sont pas symétriques.

Soit maintenant C une matrice symétrique définie positive et $C = LL^T$ sa factorisation de Cholesky (qui existe). Un moyen de préserver la symétrie est de considérer le système préconditionné

$$L^T AL(L^{-1}x) = L^T b,$$

qui s'écrit

$$By = c,$$

avec $B = L^T AL$ et $y = L^{-1}x,\ c = L^T b$. La matrice B est symétrique définie positive donc il est possible d'appliquer la méthode du Gradient Conjugué au système $By = c$.

Maintenant, par un changement de variable, il est facile d'écrire l'algorithme du Gradient Conjugué appliqué à $By = c$ sous la forme suivante :

ALGORITHM 2: PCG

* Initialisation ;
choisir x_0 ;
$r_0 = b - Ax_0$;
$z_0 = Cr_0$;
$p_0 = z_0$;
$\rho_0 = r_0^T z_0$;
* Iterations ;
for $k = 0, 1, \ldots$ **until** convergence **do**
  $q_k = Ap_k$;
  $\alpha_k = \frac{\rho_k}{p_k^T q_k}$;
  $x_{k+1} = x_k + \alpha_k p_k$;
  $r_{k+1} = r_k - \alpha_k q_k$;
  $z_{k+1} = Cr_{k+1}$;
  $\rho_{k+1} = r_{k+1}^T z_{k+1}$;
  $\beta_k = \frac{\rho_{k+1}}{\rho_k}$;
  $p_{k+1} = z_{k+1} + \beta_k p_k$;
end do

La méthode du Gradient Conjugué est définie par la condition de Galerkin liée au produit scalaire euclidien. Mais il est également possible de définir la même condition en utilisant un autre produit scalaire.

Puisque la matrice C est symétrique définie positive, elle définit le produit scalaire

$$< x, y >_C = x^T Cy \tag{18}$$

Maintenant, la matrice AC est auto-adjointe définie positive pour le produit scalaire (18). Donc il est possible d'appliquer l'algorithme du Gradient Conjugué avec le produit scalaire (18) au système préconditionné $AC(C^{-1}x) = b$.

Il est facile de voir que l'algorithme obtenu est identique à l'algorithme 2.

L'algorithme 2 est appelé l'algorithme du Gradient Conjugué Préconditionné par C.

Des hypothèses plus générales sur le préconditionnement C sont étudiées dans [1, 3].

Le coût de chaque itération est $N+M+10n$, où M est le nombre d'opérations du produit par C (en fait, il s'agit en général d'une résolution car C est souvent l'inverse d'une matrice).

4 Propriétés des méthodes de projection

Nous allons maintenant étudier les propriétés des méthodes de projection de Krylov et retrouver celles du Gradient Conjugué. Cette partie nous permettra de définir et de caractériser les méthodes de projection de Krylov dans le cas symétrique indéfini et dans le cas non symétrique.

Définition 4.1 *Pour x_0 donné, une méthode de projection de Krylov échoue à l'étape k si la condition de Galerkin (7) n'a pas une solution unique.*

Théorème 4.1 *Soit V_k une base orthonormée de l'espace de Krylov $\mathcal{K}_k(A, r_0)$ et soit $C_k = V_k^T B V_k$. La méthode de projection de Krylov n'échoue pas à l'étape k pour tout x_0 si et seulement si C_k est non singulière.*

Si la méthode n'échoue pas à l'étape k, alors

$$e_k = P_k e_0 \text{ où } P_k = I - V_k C_k^{-1} V_k^T B$$

est la matrice de la projection sur $(B^T \mathcal{K}_k(A, r_0))^\perp$ parallèlement à $\mathcal{K}_k(A, r_0)$.

L'itérée x_k est définie par $x_k = x_0 + V_k y$ où y est solution du système linéaire

$$C_k y = V_k^T B e_0. \tag{19}$$

Si la méthode n'a pas échoué jusqu'à l'étape k et si $\mathcal{K}_{k+1}(A, r_0) = \mathcal{K}_k(A, r_0)$, alors la méthode a convergé, donc $r_k = e_k = 0$.

Tant que la méthode n'échoue pas et ne converge pas, alors $dim(\mathcal{K}_k(A, r_0)) = k$.

Si la méthode n'échoue pas, alors elle converge en au plus n itérations.

Preuve. La condition d'espace s'écrit $x_k = x_0 + V_k y$ et la condition de Galerkin s'écrit $V_k^T B e_k = 0$, soit $V_k^T B e_0 - C_k y = 0$. Ce système linéaire a une solution unique pour tout e_0 si et seulement si C_k est non singulière.

Si C_k est inversible alors

$$\begin{aligned} y &= C_k^{-1} V_k^T B e_0, \\ e_k &= e_0 - V_k y = (I - V_k C_k^{-1} V_k^T B) e_0 = P_k e_0. \end{aligned}$$

Si $\mathcal{K}_{k+1}(A, r_0) = \mathcal{K}_k(A, r_0)$, alors $A^k r_0 \in \mathcal{K}_k(A, r_0)$ et $A^{-1} r_0 \in \mathcal{K}_k(A, r_0)$, donc $x^* = x_0 + A^{-1} r_0$ est solution de la condition de Galerkin. Si la méthode n'a pas échoué, cette solution est unique donc $x_k = x^*$.

La suite des espaces de Krylov est croissante, donc elle est stationnaire à partir d'un certain rang $k \leq n$ et et dans ce cas la méthode a convergé. $\diamond$

Théorème 4.2 *Si la matrice B est définie, la méthode de projection de Krylov associée n'échoue pas pour tout x_0.*

Si B n'est pas définie, il existe x_0 tel que la méthode de projection de Krylov associée échoue.

Si B est symétrique définie positive, alors la convergence est monotone : $\|e_k\|_B \leq \|e_{k-1}\|_B$.

Si de plus, BA^{-1} est définie, alors la convergence est strictement monotone : il existe $\epsilon > 0$, tel que pour tout k,

$$\|e_k\|_B \leq (1 - \epsilon)\|e_{k-1}\|_B.$$

Preuve. Si B est définie alors $\forall z \neq 0,\ V_k z \neq 0,\ (V_k z)^T B (V_k z) \neq 0$ càd $z^T C_k z \neq 0$ donc $C_k z \neq 0$ et C_k est inversible.

Si B n'est pas définie, soit $z \neq 0$ tel que $z^T B z = 0$ et soit $r_0 = z$ et $V_1 = \{r_0/\|r_0\|\}$. Alors $C_1 = 0$ et le système (19) n'a pas une solution unique (il peut avoir une infinité de solutions si $r_0^T B A r_0 = 0$).

Si B est symétrique définie positive, la condition de Galerkin est aussi une condition de minimisation :

$$\|e_k\|_B = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|x - x^*\|_B.$$

Il suffit d'appliquer cette condition à x_{k-1} pour obtenir la convergence monotone.

La preuve de la convergence strictement monotone est faite dans [12]. $\diamond$

Remarque 4.1 *Pour la méthode GC, $B = A$ donc B est symétrique définie positive et $BA^{-1} = I$ est définie. On retrouve que la méthode n'échoue pas et a une convergence strictement monotone.*

Dans la méthode GC, il est possible de construire une base $A-$orthonormée de l'espace de Krylov à l'aide d'une récurrence courte. Les théorèmes prouvés dans [5, 6] donnent une caractérisation des méthodes pour lesquelles une telle récurrence existe. Nous donnons ici une condition suffisante.

Théorème 4.3 *Si B et BA sont symétriques, alors il existe une récurrence courte permettant de calculer une base "$B-$orthogonale" de l'espace de Krylov.*

Preuve. Nous faisons la preuve par récurrence. Pour $k = 0$, le choix $p_0 = r_0$ convient. Supposons qu'il existe une base $B-$orthogonale $(p_0, \ldots, p_{k-1})$ de $\mathcal{K}_k(A, r_0)$. Nous cherchons p_k sous la forme

$$p_k = r_k + \sum_{i=1}^{k-1} \beta_i p_i.$$

Puisque B est symétrique, $(Bp_k)^T p_j = p_k^T (Bp_j)$. Alors

$p_k^T B p_j = r_k^T B p_j + \sum_{i=1}^{k-1} \beta_j p_i^T B p_j$,
or $p_i^T B p_j = 0,\ j \leq k-2, j \neq i,$ par récurrence,
et $r_k^T B p_j = (Ae_k)^T B p_j = e_k^T (BA)^T p_j = e_k^T B(Ap_j)$, car B et BA sont symétriques,
or $Ap_j \in \mathcal{K}_k(A, r_0),\ j \leq k-2$,
donc $r_k^T B p_j = 0,\ j \leq k-2$,
d'où $\beta_j = 0,\ j \leq k-2$,
et $p_k = r_k + \beta_{k-1} p_{k-1}$.

$\diamond$

5 Cas où A est symétrique indéfinie

Lorsque A est symétrique mais indéfinie, on peut choisir $B = A$ pour définir une méthode symétrique mais on ne garantit pas l'absence d'échec ou choisir $B = A^2$ qui est symétrique définie positive (donc pas d'échec et convergence monotone due à une propriété de minimisation). Dans les deux cas, il est possible d'utiliser la méthode de Lanczos puisque A est symétrique. Il existe plusieurs méthodes, détaillées par exemple dans [2] ; les méthodes SYMMLQ et MINRES sont dues à [15].

5.1 Méthode de Lanczos

Le procédé de Lanczos construit une base orthonormée V_k de l'espace de Krylov $\mathcal{K}_k(A, r_0)$. Soit $v_1 = r_0/\|r_0\|_2$, on a

$$AV_k = V_k T_k + \delta_k v_{k+1} u_k^T, \tag{20}$$

où $T_k \in \mathbb{R}^{k,k}$ est une matrice tridiagonale symétrique. On peut aussi l'écrire

$$AV_k = V_{k+1} \overline{T}_k \tag{21}$$

où

$$\overline{T}_k = \begin{pmatrix} T_k \\ \delta_k u_k^T \end{pmatrix}.$$

Remarque 5.1 *La méthode de Lanczos construit la même suite (x_k) que le gradient conjugué, mais elle le fait à partir de la relation :*

$$x_k \quad = \quad x_0 + \|r_0\| V_k {T_k}^{-1} e_1. \tag{22}$$

L'itéré n'est pas défini à l'étape k lorsque T_k est singulière (voir le théorème 4.1), ce qui peut arriver puisque la matrice A n'est pas supposée être définie. Le Gradient Conjugué, qui revient à factoriser au fur et à mesure la matrice T_k, s'arrête donc à la même étape. En fait, le procédé de Lanczos peut malgré tout être poursuivi et donc l'itéré pourra peut-être être construit à une étape ultérieure. Ce dépassement de l'obstacle n'est pas possible avec l'algorithme du Gradient Conjugué.

5.2 Méthode MINRES

On a $e_k^T A^2 y = r_k^T A y$, donc pour $B = A^2$, la condition de Galerkin s'écrit, d'après la proposition 2.4,

$$\min \|r_k\|_2.$$

La condition d'espace s'écrit $x_k = x_0 + V_k y$ d'où

$$r_k = r_0 - AV_k y = \|r_0\|_2 v_1 - V_{k+1}\overline{T}_k y = V_{k+1}(\|r_0\|_2 u_1 - \overline{T}_k y)$$

et la condition de Galerkin devient

$$\min_{y\in\mathbb{R}^k} \|\|r_0\|_2 u_1 - \overline{T}_k y\| \tag{23}$$

La méthode MINRES résout (23) en factorisant $\overline{T}_k$ à l'aide de rotations de Givens. Cette factorisation peut s'effectuer à l'aide de récurrences courtes, car T_k est symétrique. La généralisation au cas non symétrique est la méthode GMRES décrite plus loin.

5.3 Méthode CR

La méthode MINRES utilise la propriété de minimisation. La méthode CR, qui est aussi basée sur $B = A^2$, utilise quant à elle la propriété d'orthogonalité. Les deux méthodes construisent la même suite d'itérés (x_k). La condition de Galerkin s'écrit $e_{k+1}^T A^T A r_j = 0,\ j \leq k$ soit

$$r_{k+1}^T A r_j = 0,\ j \leq k, \tag{24}$$

autrement dit, les résidus sont $A-$conjugués. Les vecteurs de descente p_k sont A^2-orthonormés donc les vecteurs Ap_k sont orthogonaux au sens classique.

La méthode CR (Conjugate Residuals) utilise les récurrences courtes comme dans le Gradient Conjugué pour garantir $r_{k+1}^T A r_k = 0$ et $(Ap_{k+1})^T Ap_k = 0$. Elle applique la méthode de Lanczos pour construire la base orthonormée $(Ap_0/\|Ap_0\|_2, \ldots, Ap_{k-1}/\|Ap_{k-1}\|_2)$ de l'espace de Krylov $A\mathcal{K}_k(A, r_0)$.

5.4 Méthode SYMMLQ

Ici, on choisit $B = A$ comme dans le Gradient Conjugué. La condition de Galerkin s'écrit donc comme dans le Gradient Conjugué

$$x_k = x_0 + V_k y,\ T_k y = \|r_0\|_2 u_1. \tag{25}$$

La méthode SYMMLQ résout le système linéaire (25) comme dans la méthode de Lanczos, et donc comme dans le Gradient Conjugué, sans propriété de minimisation, puisque la matrice A n'est pas supposée être définie.

Tant qu'il n'y a pas d'échec, si le procédé de Lanczos s'arrête, la méthode SYMMLQ a convergé. En effet, les espaces de Krylov deviennent stationnaires et on peut appliquer le théorème 4.1. C'est un cas d'échec "heureux".

Il peut y avoir échec "sérieux" si la matrice T_k est singulière. En effet, dans le théorème 4.1, la matrice $C_k = V_k^T A V_k$ vaut T_k. La méthode SYMMLQ utilise une factorisation LQ de T_k pour éviter les situations d'échec.

6 Cas où A est non symétrique - méthodes de type gradient conjugué

Si A est non symétrique, on peut préconditionner le système (1) pour se ramener à un système symétrique défini positif. C'est l'idée des méthodes dites des équations normales. Voir par exemple [20, 1, 2].

6.1 Méthode CGNR

En préconditionnant à gauche par A^T, on obtient

$$A^T Ax = A^T b, \tag{26}$$

sur lequel on peut appliquer l'algorithme du Gradient Conjugué. La méthode calcule alors

$$x_k \in x_0 + \mathcal{K}_k(A^T A, A^T r_0)$$

tel que

$$\|r_k\|_2 = \min_{y \in x_0 + \mathcal{K}_k(A^T A, A^T r_0)} \|b - Ay\|_2.$$

On obtient ainsi la méthode dite CGNR, Gradient Conjugué appliqué aux équations normales qui minimise la norme euclidienne du résidu.

Une variante de CGNR, appelée LSQR, qui est très souvent utilisée pour résoudre les problèmes aux moindres carrés, est décrite dans [16].

6.2 Méthode CGNE

En préconditionnant à droite par A^T, on obtient

$$AA^T(A^{-T}x) = b, \tag{27}$$

sur lequel on peut appliquer l'algorithme du Gradient Conjugué. La méthode calcule alors

$$x_k \in x_0 + A^T \mathcal{K}_k(AA^T, r_0)$$

tel que

$$\|e_k\|_2 = \min_{y \in x_0 + A^T \mathcal{K}_k(AA^T, r_0)} \|x^* - y\|_2.$$

On obtient ainsi la méthode dite CGNE, Gradient Conjugué appliqué aux équations normales qui minimise la norme euclidienne de l'erreur.

6.3 Convergence de CGNR et CGNE

Les méthodes CGNR et CGNE ont les avantages de la méthode du Gradient Conjugué : récurrence courte, minimisation, convergence strictement monotone.

Par contre, dans la borne d'erreur (17), le conditionnement est celui de la matrice préconditionnée c'est-à-dire

$$\kappa(A^T A) = \kappa(AA^T) = \kappa(A)^2 = (\frac{\sigma_n}{\sigma_1})^2,$$

où $0 < \sigma_1 < \ldots < \sigma_n$ sont les valeurs singulières de A.

7 Cas où A est non symétrique - méthode GMRES

Une autre solution pour le cas non symétrique, comme pour le cas symétrique indéfini, est de choisir $B = A^T A$ qui est symétrique définie positive. La méthode de Krylov associée est la méthode GMRES [21].

Théorème 7.1 *La méthode GMRES est caractérisée par*

$$\begin{aligned} &x_k \in x_0 + \mathcal{K}_k(A, r_0), \\ &r_k \perp A\mathcal{K}_k(A, r_0), \end{aligned}$$

et cette condition est équivalente à

$$\|r_k\|_2 = \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|_2. \tag{28}$$

La méthode GMRES n'échoue pas (le problème (28) a une solution unique). De plus, la convergence est monotone et la solution est atteinte en au plus n itérations.

Preuve. Il suffit d'appliquer le théorème 4.2. ◇

Remarque 7.1 *Par contre, la matrice $BA = A^T A^2$ n'est pas symétrique et il n'est pas possible d'appliquer le théorème 4.3 pour définir une récurrence courte.*

Puisque la matrice $BA^{-1} = A^T$ n'est pas définie en général, la convergence peut ne pas être strictement monotone. Dans l'exemple ci-dessous, le résidu stagne jusqu'à l'itération n où il s'annule.

Exemple 7.1 *Soit $(u_1, \ldots, u_n)$ la base canonique de $\mathbb{R}^n$ et soit la matrice A définie par*

$$\begin{cases} Au_i = u_{i+1}, \ 1 \leq i \leq n-1, \\ Au_n = u_1. \end{cases}$$

Soit $b = u_1$. Le système linéaire $Ax = b$ a pour solution $x^* = u_n$.

Avec $x_0 = 0$, la méthode GMRES donne les résultats suivants :

$$\begin{cases} \mathcal{K}(A, r_0) = eng\{u_1, \ldots, u_k\}, \ 1 \leq k \leq n-1, V_{k+1} = \{u_1, \ldots, u_k\}, \ 1 \leq k \leq n-1, x_k = 0, \ r_k = u_1, \ 1 \leq k \leq \\ x_n = u_n, \ r_n = 0. \end{cases}$$

Il y a stagnation durant $n-1$ itérations et convergence vers la solution exacte à la n^{ieme} itération.

La proposition suivante caractérise les situations de stagnation dans GMRES.

Proposition 7.1 *Si $\|r_{k+1}\|_2 = \|r_k\|_2$ alors $r_k^* A r_k = 0$.*

Réciproquement, si $r_k^* A r_k = 0$, alors $r_{k+1} = r_k$ ou $r_k = r_{k-1}$.

Preuve. Si $\|r_{k+1}\|_2 = \|r_k\|_2$, alors r_k est l'unique solution du problème (28) pour l'indice $k+1$ donc $r_k = r_{k+1}$ et $r_k \perp A\mathcal{K}_{k+1}(A, r_0)$; or $r_k \in \mathcal{K}_{k+1}(A, r_0)$ d'où $r_k \perp Ar_k$.

Réciproquement, si $r_k = 0$, alors $r_{k+1} = r_k = 0$. On peut donc supposer que $r_k \neq 0$. Alors l'espace de Krylov $\mathcal{K}_{k+1}(A, r_0)$ est de dimension $k+1$.

Supposons que $r_k^* A r_k = 0$.

On sait que $r_k \in \mathcal{K}_{k+1}(A, r_0)$, alors analysons deux situations possibles.

Si r_k n'est pas dans $\mathcal{K}_k(A, r_0)$ alors $eng\{\mathcal{K}_k(A, r_0), r_k\} = \mathcal{K}_{k+1}(A, r_0)$. Comme $r_k \perp A\mathcal{K}_k$ et $r_k \perp Ar_k$, on en déduit que $r_k \perp A\mathcal{K}_{k+1}$ donc que r_k est solution du problème (28) pour l'indice $k+1$ et par unicité, $r_{k+1} = r_k$.

Sinon, $r_k \in \mathcal{K}_k$ donc $x_k - x_0 \in \mathcal{K}_k, A(x_k - x_0) \in \mathcal{K}_k$ d'où, grâce à la proposition (2.3), $x_k - x_0 \in \mathcal{K}_{k-1}$. Par conséquent, r_k est solution du problème (28) à l'indice $k-1$ et par unicité, $r_k = r_{k-1}$. $\diamond$

7.1 Lien avec la méthode d'Arnoldi

Dans le cas symétrique, les méthodes GC et MINRES sont liées au procédé de Lanczos. Ici, la méthode GMRES est liée au procédé d'Arnoldi.

Définition 7.1 *Tant que la dimension de l'espace de Krylov est maximale, le procédé d'Arnoldi construit une base orthonormée $V_{k+1} = (v_1, \ldots, v_{k+1})$ de l'espace de Krylov $\mathcal{K}_{k+1}(A, v_1)$. Cette base vérifie*

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} u_k^T = V_{k+1} \overline{H}_k \tag{29}$$

où $H_k \in \mathbb{R}^{k\times k}$ est une matrice de Hessenberg, $\overline{H}_k \in \mathbb{R}^{k+1\times k}$ est définie par

$$\overline{H}_k = \begin{pmatrix} H_k \\ h_{k+1,k} u_k^T \end{pmatrix}$$

et où $u_k^T = (0 \ldots 0\ 1) \in \mathbb{R}^k$.

La méthode GMRES peut maintenant être construite grâce à la méthode d'Arnoldi.

Théorème 7.2 *Si la méthode GMRES n'a pas convergé, le problème (28) est équivalent au problème*

$$\min_{y\in\mathbb{R}^k} (\|r_0\|_2 u_1 - \overline{H}_k y) \tag{30}$$

où V_{k+1} et $\overline{H}_k$ sont la base et la matrice construites par le procédé d'Arnoldi appliqué à $v_1 = \frac{r_0}{\|r_0\|_2}$.

Preuve. Le système (19) est équivalent au problème (28). La matrice $C_k = V_k^T B V_k$ vaut ici $(AV_k)^T(AV_k) = \overline{H}_k^T \overline{H}_k$ car V_{k+1} est un système orthonormé.

Le second membre du système (19) vaut $V_k^T B e_0 = \overline{H}_k^T V_{k+1}^T r_0 = \|r_0\|_2 \overline{H}_k^T u_1$ car v_1 est orthogonal à $v_i, i \geq 2$.

Résoudre le système (19) équivaut donc à résoudre le système

$$\overline{H}_k^T \overline{H}_k y = \|r_0\|_2 \overline{H}_k^T u_1,$$

qui est le système des équations normales associé au problème aux moindres carrés (30). ◇

Il reste à choisir une méthode de résolution de (30). La méthode GMRES, telle qu'elle est définie dans [21] et couramment utilisée, utilise une factorisation QR de la matrice $\overline{H}_k$ par des rotations de Givens. Il est alors possible de calculer $\|r_k\|_2$ sans calculer x_k et d'avoir un critère d'arrêt simple.

7.2 Convergence de GMRES

Comme pour GC, la propriété de minimisation de GMRES peut se traduire sous forme polynomiale.

Théorème 7.3 *Si A est diagonalisable, soit $A = U\Sigma U^{-1}$, où les colonnes de U forment une base de vecteurs propres et où $\Sigma = diag(\lambda_1, \ldots, \lambda_n)$ et soit $\kappa(U) = \|U\|_2 \|U^{-1}\|_2$ le conditionnement de U.*

Les itérations de GMRES vérifient

$$\|r_k\|_2 \leq \|r_0\|_2 \ \kappa(U) \ \min_{q \in \mathcal{P}_k^0} \max_{z \in \sigma(A)} |q(z)|. \tag{31}$$

Preuve. La propriété de minimisation (28) est équivalente à la propriété

$$\|r_k\|_2 = \min_{q \in \mathcal{P}_k^0} \|q(A) r_0\|_2.$$

Soit $q \in \mathcal{P}_k^0$, alors $q(A) = U q(\Sigma) U^{-1}$; soit $r_0 = U\mu$, alors $q(A) r_0 = U q(\Sigma) \mu$.

Donc $\|q(A) r_0\|_2 \leq \|U\|_2 \|q(\Sigma)\|_2 \|\mu\|_2$.

Or $\mu = U^{-1} r_0$ d'où $\|\mu\|_2 \leq \|U^{-1}\|_2 \|r_0\|_2$.

En outre $\|q(\Sigma)\|_2 = \max_{\lambda_i} |q(\lambda_i)|$, ce qui donne l'inégalité (31).

◇

7.3 Redémarrage de GMRES

L'inconvénient principal de GMRES est l'absence de récurrence. Le procédé d'Arnoldi requiert le stockage des vecteurs v_k et l'orthogonalisation du vecteur Av_{k+1} par le procédé de Gram-Schmidt modifié. Le nombre d'opérations, outre le produit par A, est donc en $O(nk^2)$. Pour limiter le coût, à la fois en mémoire et en temps de calcul, on utilise en pratique un redémarrage.

La méthode GMRES(m) effectue des cycles de m itérations de GMRES, en redémarrant avec la dernière approximation x_m. Cela permet de limiter le stockage à $O(m)$ vecteurs et de réduire le temps de calcul. Toutefois, le choix de m est délicat.

Remarque 7.2 *Les normes euclidiennes des résidus dans GMRES(m) décroissent mais il peut y avoir stagnation. Dans l'exemple 7.1, la méthode GMRES(m) stagne sans converger pour tout $m < n$.*

Voici un squelette de l'algorithme GMRES(m), où n'est pas détaillée la factorisation QR de la matrice $\overline{H}$ avec des rotations de Givens.

ALGORITHM 3: GMRES(m)

* Initialisation ;
choisir x_0 ;
$r_0 = b - Ax_0$;
* Iterations ;
until convergence **do**
  $v_1 = \frac{r_0}{\|r_0\|_2}$;
  * procédé d'Arnoldi ;
  for $j = 1, m$
    $w = Av_j$;
    for $i = 1, j$
      $h_{ij} = v_i^T w$;
      $w = w - h_{ij} v_i$;
    end for ;
    $h_{j+1,j} = \|w\|_2$;
    $v_{j+1} = w / h_{j+1,j}$;
    * problème aux moindres carrés
    $\overline{H}_j = Q_j R_j$;
    calculer $\|r_j\|_2$;
    test de convergence
  end for ;
  calculer y_m solution de $\min_y (\|r_0\|_2 e_1 - \overline{H}_m y)$;
  $x_m = x_0 + V_m y_m$;
  $r_m = b - Ax_m$;
  test de convergence
  $x_0 = x_m$; $r_0 = r_m$;
end do

Dans un cycle de m itérations de GMRES(m), le nombre d'opérations vaut $2nm^2 + mN + O(nm)$ et il faut stocker les $m + 1$ vecteurs v_j.

8 Cas où A est non symétrique - méthodes de gradient bi-conjugué

Nous venons de voir que la méthode GMRES a de bonnes propriétés de minimisation mais nécessite un redémarrage car elle ne possède pas de récurrence courte. A l'opposé, les méthodes de type gradient bi-conjugué utilisent une récurrence courte mais n'ont pas de propriété de minimisation et sont susceptibles d'échouer. La méthode BICG est une méthode de projection de Krylov, les variantes CGS, BICGSTAB et QMR sont des méthodes polynomiales mais ne sont plus des méthodes de projection. Alors que la méthode GMRES est liée au procédé d'Arnoldi, les méthodes bi-conjuguées sont connectées à la méthode de Lanczos non symétrique. Pour plus de détails sur ces méthodes, voir par exemple [20, 2, 3, 14, 11].

8.1 Construction de BICG

La méthode du gradient bi-conjugué (BICG) résout le système augmenté

$$\begin{pmatrix} A & 0 \\ 0 & A^T \end{pmatrix} \begin{pmatrix} x \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} b \\ \tilde{b} \end{pmatrix}$$

par la méthode de projection de Krylov où la matrice B est choisie égale à

$$\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}$$

Proposition 8.1 *La méthode BICG est définie par les choix de x_0 et de $\tilde{x}_0$ et par les conditions d'espace et de Galerkin suivantes :*

$$\begin{aligned} &x_k \in x_0 + \mathcal{K}_k(A, r_0), \\ &\tilde{x}_k \in \tilde{x}_0 + \mathcal{K}_k(A^T, \tilde{r}_0), \\ &r_k \perp \mathcal{K}_k(A^T, \tilde{r}_0), \\ &\tilde{r}_k \perp \mathcal{K}_k(A, r_0). \end{aligned}$$

La méthode risque d'échouer.

La méthode BICG possède une récurrence courte.

Preuve. Soit

$$\tilde{A} = \begin{pmatrix} A & 0 \\ 0 & A^T \end{pmatrix}.$$

L'espace de Krylov associé est

$$\begin{pmatrix} \mathcal{K}_k(A, r_0) \\ \mathcal{K}_k(A^T, \tilde{r}_0) \end{pmatrix},$$

ce qui donne bien la condition d'espace. De plus,

$$B\tilde{A}^{-1} = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix},$$

ce qui donne bien la condition de Galerkin.

La matrice B n'est pas définie donc la méthode risque d'échouer, d'après le théorème 4.2.

Les matrices B et $B\tilde{A}$ sont symétriques, la méthode possède donc une récurrence courte, d'après le théorème 4.3. ◇

L'algorithme BICG est construit de la façon suivante.

ALGORITHM 4: BICG

* Initialisation ;
choisir x_0 et $\tilde{b}$ et $\tilde{x}_0$;
$r_0 = b - Ax_0$; $\tilde{r}_0 = \tilde{b} - A\tilde{x}_0$;
$p_0 = r_0$; $\tilde{p}_0 = \tilde{r}_0$;
* Iterations ;
for $k = 0, 1, \ldots$ **until** convergence **do**
$\alpha_k = \frac{\tilde{r}_k^T r_k}{\tilde{p}_k^T A p_k}$;
$x_{k+1} = x_k + \alpha_k p_k$;
$r_{k+1} = r_k - \alpha_k A p_k$;
$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k A^T \tilde{p}_k$;
$\beta_{k+1} = \frac{\tilde{r}_{k+1}^T r_{k+1}}{\tilde{r}_k^T r_k}$;
$p_{k+1} = r_{k+1} + \beta_{k+1} p_k$;
$\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_{k+1} \tilde{p}_k$;
end do

Les résidus et les directions de descente vérifient les conditions d'orthogonalité suivantes :

$$r_{k+1}^T \tilde{r}_k = 0, \ A p_{k+1}^T \tilde{p}_k = p_{k+1}^T A^T \tilde{p}_k = 0.$$

8.2 Lien avec Lanczos non symétrique

Théorème 8.1 *La méthode BICG construit deux bases $V_k = (v_1, \ldots, v_k)$ et $W_k = (w_1, \ldots, w_k)$ des espaces de Krylov $\mathcal{K}_k(A, r_0)$ et $\mathcal{K}_k(A^T, \tilde{r}_0)$ qui vérifient*

$$\begin{aligned} AV_k &= V_k T_k + \delta_k v_{k+1} u_k^T, \\ A^T W_k &= W_k T_k^T + \gamma_k w_{k+1} u_k^T, \\ V_k^T W_k &= I, \end{aligned} \tag{32}$$

où $T_k \in \mathbb{R}^{k \times k}$ est une matrice tridiagonale et où $u_k^T = (0 \ldots 0 \ 1) \in \mathbb{R}^k$.

Autrement dit, la méthode BIGC applique la méthode de Lanczos non symétrique, appelée aussi dans la suite méthode Bi-Lanczos, aux vecteurs de départ $v_1 = r_0/\|r_0\|_2$ et $w_1 = \mu \tilde{r}_0$ tel que $v_1^T w_1 = 1$.

La méthode BICG résout le système linéaire

$$T_k y = \|r_0\|_2 u_1.$$

Elle échoue si T_k est singulière.

Preuve. La preuve est la même que pour CG, où V_k et W_k sont les bases $(r_0, \ldots, r_{k-1})$ et $(\tilde{r}_0, \ldots, \tilde{r}_{k-1})$ normalisées de manière à avoir $V_k^T W_k = I$. Il y a échec si la matrice

$$C_k = \begin{pmatrix} 0 & T_k^T \\ T_k & 0 \end{pmatrix},$$

du théorème 4.2 est singulière, donc si la matrice T_k est singulière.

◇

Il existe des situations d'échec "heureux" dans la méthode Bi-Lanczos, comme pour les méthodes de Lanczos et d'Arnoldi.

Proposition 8.2 *Si la méthode Bi-Lanczos s'est poursuivie jusqu'à l'indice $k-1$ et s'arrête parce que $Av_k \in eng\{v_k, v_{k-1}\}$ (dans ce cas, $Aw_k \in eng\{w_k, w_{k-1}\}$), alors $\mathcal{K}_k(A, v_1) = \mathcal{K}_{k+1}(A, v_1)$ et $\mathcal{K}_k(A^T, w_1) = \mathcal{K}_{k+1}(A^T, w_1)$.*

Preuve. Par hypothèse, il n'y pas eu d'échec auparavant et on peut appliquer le théorème 4.1. ◇

Cet arrêt est un échec "heureux" pour la méthode BICG puisqu'alors la méthode a convergé.

8.3 Convergence de BICG

La méthode BICG présente l'avantage de posséder une récurrence courte. Par contre, elle requiert à chaque itération deux produits matrice-vecteur, par A et A^T. De plus, elle peut échouer (échec "sérieux") si la matrice T_k est singulière. Enfin, le résidu ou l'erreur ne vérifient pas de propriété de minimisation, on ne peut pas prouver une convergence monotone. On observe effectivement dans de nombreux cas une convergence très irrégulière.

Pour éviter les échecs dans la méthode Bi-Lanczos, il est possible d'utiliser une version dite "look-ahead" [17, 8]. L'idée est de construire plusieurs vecteurs à la fois qui sont bi-orthogonaux par blocs. Cette version est efficace sauf dans les cas d'échecs dits "incurables".

8.4 Variantes CGS et BICGSTAB

Pour éviter les produits par A^T, il est possible de modifier le polynôme sous-jacent dans BICG. C'est l'idée des méthodes CGS [23] et BICGSTAB [25]. Il est à noter que ces variantes de méthodes de Krylov ne sont plus des méthodes de projection mais seulement des méthodes polynomiales.

9 Cas où A est non symétrique - méthode QMR

La méthode QMR permet d'éviter une convergence irrégulière. Cette méthode polynomiale n'est plus à proprement parler une méthode de projection de Krylov. Elle est basée aussi sur la méthode Bi-Lanczos et impose une condition de quasi-minimisation [9, 7]. Plus précisément, on a

$$\begin{aligned} x_k &= x_0 + V_k y, \\ r_k &= r_0 - AV_k y = V_{k+1}(\beta u_1 - \overline{T}_k y). \end{aligned}$$

Puisque V_{k+1} n'est pas un système orthogonal, minimiser $\|r_k\|$ serait trop coûteux. On définit le problème de quasi-minimisation

$$\min_y \|\Omega_{k+1}^{-1}(\gamma u_1 - \Omega_{k+1}\overline{T}_k y)\|_2 \tag{33}$$

où Ω_{k+1} est une matrice diagonale.

Ce problème est résolu par une factorisation QR de la matrice $\Omega_{k+1}\overline{T}_k$.

La méthode de Lanczos avec "look-ahead" permet d'éviter les situations d'échec. Une version "Transpose-Free" (TFQMR) ne requiert pas le produit par A^T. La convergence n'est pas monotone mais est plus régulière que pour les variantes de BICG. En outre, il existe un résultat de convergence.

10 Problèmes numériques dans les méthodes de Krylov

10.1 Perte d'orthogonalité et dérive du résidu

Les méthodes de Krylov reposent sur des conditions d'orthogonalité, qui sont souvent vérifiées par récurrence. De même, l'itéré x_k et le résidu r_k sont souvent calculés par deux récurrences, qui vérifient implicitement $r_k = b - Ax_k$. Ces conditions sont démontrées en arithmétique exacte, mais ne sont pas vérifiées en arithmétique flottante, à cause des erreurs d'arrondi générées. Cela se traduit par deux phénomènes [] :

- une perte d'orthogonalité ; par exemple, dans le Gradient Conjugué, les directions de descente ne sont plus conjuguées et les résidus ne sont plus orthogonaux. Cette perte d'orthogonalité engendre une irrégularité et un ralentissement de la convergence. Dans les versions matrix-free notamment, le produit par A engendre des erreurs d'arrondi assez grandes pour provoquer une perte d'orthogonalité. Celle-ci peut être corrigée par une réorthogonalisation totale ou partielle.

- un résidu calculé qui n'est plus égal à $b-Ax$; par exemple, dans le Gradient Bi-Conjugué, on observe une dérive entre r_k calculé et $b - Ax_k$. Par conséquent, le critère d'arrêt basé sur r_k n'est plus valide et les itérations risquent de s'arrêter avant réelle convergence. Ce problème peut être corrigé en recalculant régulièrement le résidu.

10.2 Breakdowns et near-breakdowns

Certaines méthodes de Krylov peuvent échouer, par exemple SYMMLQ ou BICG, lorsque la matrice tridiagonale du procédé de Lanczos ou de Bi-Lanczos devient singulière. De même, le procédé de Lanczos peut s'arrêter si l'espace de Krylov devient invariant (échec "heureux"). Numériquement, la situation se dégrade dès que la matrice de Lanczos devient proche de la singularité ou dès que l'espace de Krylov devient presque invariant. On parle alors de "near-breakdown".

Il faut en fait appliquer la méthode de Lanczos avec "look-ahead" dès qu'un problème numérique est détecté [].

11 Préconditionnement

En général, les méthodes de Krylov convergent trop lentement et il faut préconditionner le système. L'idée est de trouver une matrice C telle que CA soit mieux conditionnée que A et telle que le produit par C soit peu coûteux. L'idéal serait de choisir $C = A^{-1}$, aussi cherche-t-on à approcher l'inverse de A. Pour le Gradient Conjugué, un préconditionnement symétrique défini positif garantit de conserver les propriétés de la méthode. Il existe différentes façons de construire un préconditionnement. Nous donnons ci-dessous un bref aperçu, plus de détails peuvent se trouver dans [20, 14].

11.1 Décomposition de A

Les préconditionnements les plus simples sont basés sur les méthodes linéaires, donc sur une décomposition de A.

Le préconditionnement diagonal, dit aussi de Jacobi, consiste à choisir $C = D^{-1}$ où $D = diag(A)$.

Le préconditionnement SSOR consiste à choisir $C = (D+\omega L)D^{-1}(D+\omega U)$ où $A = D+L+U$, avec D diagonale, L triangulaire inférieure et U triangulaire supérieure. En général, on choisit $\omega = 1$.

Ces deux préconditionnements sont symétriques définis positifs dès que A l'est.

11.2 Factorisation incomplète

Les méthodes itératives sont souvent appliquées à des matrices creuses, dont une grande partie des coefficients sont nuls. Un des inconvénients des méthodes directes appliquées aux matrices creuses est le coût de stockage induit par le remplissage lors de la factorisation de Gauss ou de Cholesky, voir par exemple [18]. L'idée des factorisations incomplètes est de limiter le remplissage. On obtient alors

$$A = LU + R$$

et le préconditionnement est défini par $C = U^{-1}L^{-1}$.

Il existe diverses stratégies pour limiter le remplissage, mais la factorisation incomplète peut ne pas exister. Toutefois, elle existe pour toute stratégie si A est une $M-$matrice [13]. Le cas le plus simple est la méthode ILU(0) où aucun remplissage n'est autorisé.

Ces préconditionnements sont très souvent utilisés pour leur efficacité, bien qu'ils requièrent un stockage supplémentaire conséquent et un temps de calcul assez important.

11.3 Préconditionnement polynomial

Dans la mesure où les méthodes de Krylov sont des méthodes polynomiales, il paraît naturel de les préconditionner par des polynômes. L'objectif est de choisir un polynôme qui approche l'inverse de A. Plus précisément, on cherche $q \in \mathcal{P}_m$ tel que $\|1 - Xq(X)\|$ soit minimal pour une norme définie sur l'espace des polynômes. Il existe principalement deux choix, une norme uniforme et une norme aux moindres carrés, toutes deux définies sur un compact contenant le spectre de A [19].

L'avantage est un coût mémoire négligeable, mais l'inconvénient est une efficacité parfois réduite.

11.4 Inverse approché

Ici, il s'agit de trouver une matrice C qui minimise $\|I - CA\|$ ou $\|I - AC\|$ pour une norme à préciser. Ce type de préconditionnement peut s'avérer très efficace, mais coûteux à calculer. En outre, les conditions d'existence sont mal définies dans le cas non symétrique.

11.5 Multigrille et multiniveaux

Les méthodes multigrilles et multiniveaux sont des méthodes itératives qui peuvent être utilisées en soi. Mais, comme pour les méthodes linéaires de type Gauss-Seidel, il est possible de définir un préconditionnement à partir de ces méthodes. Par exemple, un $V-$cycle d'une méthode multigrille est un préconditionnement symétrique défini positif si A l'est.

L'intérêt de ces méthodes est de réduire notablement le conditionnement de A, lorsque A est issue d'une discrétisation d'EDP. Si h est le pas de discrétisation, le conditionnement passe souvent de $O(h^{-1})$ à $O(1)$.

11.6 Problèmes approchés

Les préconditionnements décrits jusqu'ici sont basés sur des concepts algébriques. Lorsque la matrice A est issue d'une discrétisation d'EDP, il peut être très efficace de construire un préconditionnement à partir d'une décomposition de l'EDP ou d'une EDP plus simple.

11.7 Déflation et systèmes augmentés

Le phénomène de convergence superlinéaire est à la base des méthodes de projection et de déflation pour accélérer la convergence d'une séquence de systèmes linéaires ou à chaque redémarrage de GMRES(m). L'idée est de calculer les valeurs de Ritz et les vecteurs de Ritz associés aux plus petites valeurs propres. Ces vecteurs engendrent un espace qui approche un espace invariant de A. Les méthodes de déflation et de systèmes augmentés définissent une projection associée à ce sous-espace [4].

References

[1] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor. A taxonomy for Conjugate Gradient Methods. *SIAM Journal on Numerical Analysis*, 26:1542–1568, 1990.

[2] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods - 2nd edition*. SIAM / netlib, Philadelphia, PA, 1994.

[3] A.M. Bruaset. *A survey of preconditioned iterative methods*. Pitman Research Notes in Mathematics Series. Longman Scientific and Technical, 1995.

[4] J. Erhel and F. Guyomarc'h. An augmented conjugate gradient method for solving consecutive symmetric positive definite systems. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1279–1299, 2000.

[5] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient methods. *SIAM Journal on numerical analysis*, 21:352–362, 1984.

[6] V. Faber and T. Manteuffel. Orthogonal error methods. *SIAM journal on numerical analysis*, 24(1):170–187, 1987.

[7] R. Freund. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM journal on scientific computing*, 14:470–482, 1993.

[8] R. Freund, M. Gutknecht, and N. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-hermitian matrices. *SIAM journal on scientific computing*, 14:137–158, 1993.

[9] R. Freund and N. Nachtigal. QMR : a quasi-minimal residual method for non-Hermitian linear systems. *Numerische mathematik*, 60:315–339, 1991.

[10] G.H Golub and C.F Van Loan. *Matrix Computations. third edition*. John Hopkins, 1996.

[11] M. Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta numerica*, 6:271–397, 1997.

[12] W. D. Joubert and T. A. Manteuffel. *Iterative Methods for Nonsymmetric Linear Systems*, chapter 10, pages 149–171. Academic Press, 1990.

[13] J. A. Meijerink and H. A. Van Der Vorst. An iterative solution method for linear systems of which the coefficientmatrix is a symmetric M-matrices. *Math. Comp.*, 31(137):148–162, 1977.

[14] G. Meurant. *Computer solution of large linear systems.* North Holland, Amsterdam, 1999.

[15] C. Paige and M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12:617–629, 1975.

[16] C. Paige and M. Saunders. LSQR : an algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8:43–71, 1982.

[17] B. Parlett, D. Taylor, and Z. Liu. A look-ahead Lanczos algorithm for unsymmetric matrices. *Mathematics of computation*, 44:105–124, 1985.

[18] J Reid, I. Duff, and A. Erisman. *Direct methods for sparse matrices.* Oxford University Press, London, 1986.

[19] Y. Saad. Practical use of polynomial preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.*, 6(4):865–881, 1985.

[20] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, 1996.

[21] Y Saad and H Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetriclinear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[22] G. L. G. Sleijpen and H. A. Van der Vorst. A jacobi-davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 17:401–425, 1996.

[23] P. Sonnefeld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM journal on scientific and statistical computing*, 10(36-52), 1989.

[24] A. van der Sluis and H. van der Vorst. the rate of convergence of conjugate gradients. *Numerische Mathematik*, 48:543–560, 1986.

[25] H. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM journal on scientific and statistical computing*, 13:631–644, 1992.

[26] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, 1962.