Chapitre 5

Résolution de systèmes linéaires par des méthodes de Krylov

5.1 Introduction

Que fait-on lorsqu'il s'agit de résoudre un système linéaire et que les méthodes basées sur des factorisations matricielles sont trop coûteuses (en temps de calcul ou en mémoire) compte tenu du matériel informatique utilisé ? On utilise des méthodes itératives qui génèrent une suite d'itérés sensés converger vers la solution du problème. Le but de ce chapitre est de présenter les méthodes qui figurent parmi les plus utilisées : les méthodes basées sur un espace dit de Krylov. Ce chapitre sera notamment l'occasion de décrire la méthode GMRES, la méthode du gradient conjugué, le but du précondiitonnement et la nécessité de disposer de bons critères d'arrêt des itérations.

Dans les cas pratiques il est bon de se rappeler qu'il ne faut se tourner vers les méthodes itératives que lorsque les méthodes directes ne sont pas utilisables, car la mise en œuvre d'une méthode itérative peut nécessiter beaucoup d'efforts, notamment concernant les techniques de préconditionnement.

5.2 Généralités

On définit par l'espace de Krylov de d'ordre m associé à la matrice carrée inversible $A \in \mathbb{R}^{n\times n}$ et $b \in \mathbb{C}^n$ par $\mathcal{K}(A, b, m) = Span\{b, Ab, ..., A^{m-1}b\}$. Il est clair que les espaces de Krylov sont des espaces emboîtés lorsque m croît. Dans ce chapitre, sauf précision contraire, $\|\|$ est la norme Euclidienne pour les vecteurs et la norme induite correspondante pour les matrices.

Proposition 5.1 *Montrez, en utilisant le polynôme caractéristique, que la solution $x = A^{-1}b$ appartient à l'espace de Krylov de d'ordre n (noté $\mathcal{K}(A, b, n)$). Noter que cet espace peut être de dimension très inférieure a n (exemple si A est la matrice identité).*

Preuve 5.1 *Si $q(t)$ est le polynôme caractéristique, on a $q(t) = \sum_{j=0}^{n} \alpha_j t^j$. Donc $\alpha_0 =$*

$q(0) = det(A) \neq 0$ ssi A est inversible. De plus, comme

$$0 = q(A) = \alpha_0 I + \alpha_1 A + ... + \alpha_n A^n, \tag{5.1}$$

on a

$$A^{-1} = -\frac{1}{\alpha_0} \sum_{j=0}^{n-1} \alpha_{j+1} A^j.$$

Ainsi $x = A^{-1}b$ appartient à l'espace de Krylov de d'ordre n associé à A et b et noté $\mathcal{K}(A, b, n) = Span\{b, Ab, ..., A^{n-1}b\}$.

□

Les méthodes de Krylov se répartissent en plusieurs classes suivant la manière dont l'itéré $x_k \in \mathcal{K}(A, b, k)$ est construit. Par convention on pose $x_0 = 0$. Si $x_0 \neq 0$, c'est a dire, si l'on dispose d'une approximation de la solution, on se ramène au cas précédent en résolvant $Az = b - Ax_0$ puis en faisant la mise à jour $x = x_0 + z$. On trouve

- *L'approche de Ritz-Galerkin : x_k est tel que $b - Ax_k \perp \mathcal{K}(A, b, k)$.*
- *Le résidu minimum : trouver $x_k \in \mathcal{K}(A, b, k)$ tel que $||b - Ax_k||_2$. est minimum*
- *L'approche de Petrov-Galerkin : trouver x_k tel que $b - Ax_k$ est orthogonal à un espace de dimension k (éventuellement différent de $\mathcal{K}(A, b, k)$).*
- *L'approche erreur minimum : trouver $x_k \in A^T\mathcal{K}(A, b, k)$ tel que $||b - Ax_k||_2$ est minimal.*

5.3 La méthode GMRES

5.3.1 Présentation de l'algorithme

Dans l'algorithme GMRES, on choisit $x_k \in \mathcal{K}(A, b, k)$ tel que $||b - Ax_k||_2$ est minimum. Soit l'algorithme suivant :

Arnoldi's algorithm

$v_1 = b/\|b\|$
For j=1,2, ... m-1 Do
  Compute $h_{ij} = v_i^T A v_j$ for $i = 1, j$
  Compute $w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i$
  $h_{j+1,j} = \|w_j\|$
  If $(h_{j+1,j} = 0)$ then Stop
  $v_{j+1} = w_j / h_{j+1,j}$
EndDo

Proposition 5.2 *Les vecteurs v_j générés par l'algorithme sont orthogonaux.*

Preuve 5.2 *Démonstration. En effet, à l'étape j de l'algorithme, on réalise l'orthogonalisation de Schmidt de Av_j par rapport à v_i, $i \leq j$, pour obtenir v_{j+1}. Les v_i, $i \leq j + 1$ sont donc bien orthogonaux.*

□

Proposition 5.3 *Si à l'étape j_s l'algorithme rencontre une quantité h_{j_s+1,j_s} nulle, il s'arrête.*

Les quantités v_j et h_{ij} générées par l'algorithme pour $j < j_s$ peuvent être réécrites à chaque pas de la boucle en j sous forme matricielle

$$AV_j = V_{j+1}\bar{H}_j,$$

où $\bar{H}_j \in \mathbb{R}^{j+1\times j}$ est une matrice de Hessenberg supérieure.

Preuve 5.3 *Démonstration. En effet, d'après les étapes 4. et 7. de l'algorithme $h_{j+1,j}v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}v_i$, ce qui s'écrit bien $AV_j = V_{j+1}\bar{H}_j$ avec $V_j = [v_1, \ldots, v_j] \in \mathbb{R}^{n\times j}$ et $\bar{H}_j = [h_{i,j}] \in \mathbb{R}^{j+1\times j}$ Hessenberg supérieure.*

□

Proposition 5.4 *On se place au dernier pas j_s de l'algorithme. On a alors $AV_{j_s} = V_{j_s}H_{j_s}$, où la matrice H_{j_s} est une matrice carrée d'ordre j_s. Les valeurs propres de H_{j_s} sont des valeurs propres de A. Si y est un vecteur propre de H_{j_s} associé a la valeur propre λ (de A et de H_{j_s}), $V_{j_s}y$ est un vecteur propre de A associé.*

Preuve 5.4 *Démonstration. Si pour $y \neq 0$, $H_{j_s}y = \lambda y$, $AV_{j_s}y = V_{j_s}H_{j_s}y = \lambda V_{j_s}y$, avec $V_{j_s}y \neq 0$. Donc toute valeur propre de H_{j_s} est une valeur propre de A. Pour tout vecteur propre y de H_{j_s}, $V_{j_s}y$ est un vecteur propre de A.*

□

Proposition 5.5 *Soit $H_j = V_j^T AV_j$. La matrice H_j est Hessenberg supérieure. En particulier, si A est symétrique, H_j est tridiagonale.*

Preuve 5.5 *Démonstration. On sait que $H_j \in \mathbb{R}^{j\times j}$ est Hessenberg supérieure (car elle est constituée des j premières lignes de la matrice rectangulaire Hessenberg supérieure $\bar{H}_j \in \mathbb{R}^{j+1\times j}$). Si de plus A est symétrique, $\bar{H}_j = V_{j+1}^T AV_j$. et $H_j^T = (V_j^T AV_j)^T = V_j^T A^T V_j = H_j$. Donc H_j est carrée Hessenberg supérieure et symétrique ; elle est donc carrée et tridiagonale.*

□

Proposition 5.6 *L'espace image de V_j, pour j inférieur à j_s, est $\mathcal{K}(A, b, j)$. L'espace $\mathcal{K}(A, b, j_s)$ est un espace invariant pour A.*

Preuve 5.6 *Démonstration. Par récurrence. Vrai pour $j = 1$. Supposons le résultat suivant vrai au rang j : il existe une matrice $X_j \in \mathbb{R}^{j\times j}$ telle que $[b, \ldots, A^{j-1}b] = [v_1, \ldots v_j]X_j$,* *la matrice X_j étant triangulaire supérieure inversible (éléments non nuls sur la diagonale). Alors , on posant $\beta = \|b\|$,*

$$\begin{aligned}
[b, \dots, A^{j-1}b, A^j b] &= [\beta v_1, A[b, \dots, A^{j-1}b]] = [\beta v_1, A[v_1, \dots v_j]X_j] \\
&= [\beta v_1, V_{j+1}\bar{H}_j X_j] = V_{j+1} \begin{bmatrix} \beta & \\ 0 & \\ \vdots & \bar{H}_j X_j \\ 0 & \end{bmatrix}.
\end{aligned}$$

Montrons que la matrice entre crochets que nous appelons X_{j+1} est triangulaire supérieure inversible. La matrice $\bar{H}_j$ est de rang j (Hessenberg avec éléments non nuls sur la sous-diagonale sinon l'algorithme se serait arrêté). La matrice $\bar{H}_j X_j$ est Hessenberg supérieure et son élément sous diagonal de la colonne k de $\bar{H}_j$ par le kième élément diagonal de X_j : il est donc non nul. La matrice X_{j+1} est donc triangulaire supérieure à éléments diagonaux non nuls : elle est inversible. Enfin on a $AV_{j_s} = V_{j_s}H_{j_s}$ donc comme les colonnes de V_{j_s} forment une base de $\mathcal{K}(A, b, j_s)$, on a $A\mathcal{K}(A, b, j_s) \subset \mathcal{K}(A, b, j_s)$.

□

Proposition 5.7 *L'itéré x_j minimisant la norme du résidu $\|b - Ax\|$ sur l'espace $\mathcal{K}(A, b, j)$ s'écrit $x_j = V_j z_j$ où z_j minimise $\|\|b\|e_1 - \bar{H}_j z_j\|$.*

Preuve 5.7 *Démonstration.* *Si x est dans l'image de V_j, il existe $z \in \mathbb{R}^j$ tel que $x = V_j z$. Alors $\|b - Ax\| = \|\|b\|v_1 - AV_j z\| = \|\|b\|v_1 - V_{j+1}\bar{H}_j z\| = \|V_{j+1}(\|b\|e_1 - \bar{H}_j z\|)$. La norme euclidienne étant unitairement invariante, $\|b - Ax\| = \|\|b\|e_1 - \bar{H}_j z\|$. Donc on est ramené à la résolution du problème de moindres carrés $\min_{z\in\mathbb{R}^j} \|\|b\|e_1 - \bar{H}_j z\|$. Soit z_j la solution obtenue. La solution du problème de départ est $V_j z_j$.*

□

Proposition 5.8 *Le pas j_s étant celui où se produit l'arrêt de l'algorithme GMRES, x_{j_s} est la solution du système linéaire $Ax = b$.*

Preuve 5.8 *Démonstration.* *En reprenant la démonstration de la question précédente, pour le pas j_s, on obtient que $\|b - Ax\| = \|\|b\|e_1 - \bar{H}_{j_s} z_{j_s}\| = \|\|b\|e_1 - \bar{H}_{j_s} z_{j_s}\|$. La matrice H_{j_s} étant carrée et inversible (les valeurs propres de H_s sont des valeurs propres de la matrice inversible A), le minimum $\|\|b\|e_1 - \bar{H}_{j_s} z_{j_s}\|$ est nul à l'optimum z_{j_s}. Donc $x_{j_s} = V_{j_s} z_{j_s}$ vérifie $\|b - Ax_{j_s}\| = \|\|b\|e_1 - \bar{H}_{j_s} z_{j_s}\| = 0$.*

□

En rassemblant les propriétés ci-dessus, nous obtenons l'algorithme GMRES :

GMRES algorithm	
1.	x_0 *initial guess*, $r_0 = b - Ax_0$, $\beta = \|r_0\|$ *and* $v_1 = r_0/\beta$
2.	*For k=1,2, ... Do*
3.	*Compute* $w_k = Av_k$
4.	*For i=1, ...,k, Do*
5.	$h_{i,k} = w_k^T v_i$
6.	$w_k = w_k - h_{i,k} v_i$
7.	*EndDo*
8.	$h_{k+1,k} = \|w_k\|$
9.	*If* $h_{k+1,k} = 0$ *set* $m = k$ *and Goto 12*
10.	$v_{k+1} = w_k / h_{k+1,k}$
11.	*endDo*
12.	*Set-up the* $(m+1) \times m$ *matrix* $\bar{H}_m = (h_{i,j})_{1 \le i \le m+1, 1 \le j \le m}$
13.	*Compute,* y_m *the solution of* $\|\beta e_1 - \bar{H}_m y\|_2$
14.	*Compute,* $x_m = x_0 + V_m y_m$

Notons que la résolution du problème de moindres carrés en 13. est réalisée par une méthode stable (Givens),

5.3.2 GMRES restarté (ou redémarré)

L'algorithme GMRES peut être lent et nécessiter un stockage trop important pour les vecteurs v_j. *C'est pour cela que l'on utilise l'algorithme redémarré suivant :*

Restarted GMRES : GMRES(m)	
1.	x_0 *initial guess*, $r_0 = b - Ax_0$, $\beta = \|r_0\|$ *and* $v_1 = r_0/\beta$
2.	*For k=1,2, ... Do*
3.	*Compute* $w_k = Av_k$
4.	*For i=1, ...,k, Do*
5.	$h_{i,k} = w_k^T v_i$
6.	$w_k = w_k - h_{i,k} v_i$
7.	*EndDo*
8.	$h_{k+1,k} = \|w_k\|$
9.	*If* $h_{k+1,k} = 0$ *set* $m = k$ *and Goto 12*
10.	$v_{k+1} = w_k / h_{k+1,k}$
11.	*endDo*
12.	*Compute,* y_m *the solution of* $\|\beta e_1 - \bar{H}_m y\|_2$
13.	*Compute,* $x_m = x_0 + V_m y_m$
14.	*If* $h_{m+1,m} \neq 0$ *then* $x_0 = x_m$ *Goto 1*

Nous étudions à présent la convergence de l'algorithme redémarré. Une première chose est que au passage à l'algorithme redémarré, on perd la propriété de terminaison en un nombre fini de pas. Il existe des conditions nécessaires et suffisantes de convergence de l'algorithme pour toute matrice, malheureusement elles font intervenir l' "image numérique" généralisé, qui est une quantité que l'on se sait pas actuellement exploiter.

Nous citons donc ici des conditions de convergence plus utilisables en pratique.

Proposition 5.9 *Soit A une matrice diagonalisable telle que $A = VDV^{-1}$, où D est diagonale. Pour l'algorithme non redémarré,*

$$\|Ax_j - b\| \leq \|V\|\|V^{-1}\| \min_{Q\in\mathcal{P}_j, Q(0)=1} \max_{\lambda\in \mathrm{sp}(A)} |Q(\lambda)|\|Ax_0 - b\|,$$

où $\mathcal{P}_j$ est l'espace vectoriel des polynômes de degré au plus j.

Preuve 5.9 Démonstration. *Toujours pour $x_0 = 0$, sans perdre de généralité, $x_j = \sum_{i=0}^{j-1} \alpha_i A^i b$ minimise $\|Ax - b\|$. Donc les α_i minimisent $\|b - A\sum_{i=0}^{j-1} \alpha_i A^i b\| = \|Q(A)b\|$, où $Q(t) = t - \sum_{i=1}^{j} \alpha_i t^i$. Ainsi,*

$$\begin{aligned}
\|Ax_j - b\| &= \min_{Q\in\mathcal{P}_j, Q(0)=1} \|Q(A)b\|, \\
&= \min_{Q\in\mathcal{P}_j, Q(0)=1} \|VQ(D)V^{-1}b\| \\
&\leq \|V\|\|V^{-1}\|\|b\| \min_{Q\in\mathcal{P}_j, Q(0)=1} \|Q(D)\| \\
&= \|V\|\|V^{-1}\| \min_{Q\in\mathcal{P}_j, Q(0)=1} \max_{\lambda\in \mathrm{sp}(A)} |Q(\lambda)|\|Ax_0 - b\|.
\end{aligned}$$

□

On définit l' image numérique *d'une matrice comme la partie (convexe, théorème de Haussdorf) du plan complexe $\mathrm{NR}(A) = \{\frac{z^H Az}{z^H z}, z \neq 0\}$. On suppose que l'image numérique de A est inclus dans un disque de centre c et de rayon r, avec $r < |c|$. Ainsi 0 ne fait pas partie de l'image numérique de A. On appelle* rayon numérique *de A la quantité $r(A) = \max\{\frac{|z^H Az|}{z^H z}, z \neq 0\}$.*

Proposition 5.10 *Pour toute matrice carrée, $r(A^m) \leq r(A)^m$.*

Preuve 5.10 Démonstration.

1. *Il suffit, quitte à considérer $A/r(A)$, de montrer que si $r(A) \leq 1$, alors $r(A^m) \leq 1$.*
2. *Soit $w_k = e^{2\pi k/m}$, $k = 1 \ldots m$ une racine mème de l'unité. Comme $1 - z^m = \prod_{k=1}^{m}(1 - w_k z)$ (considérer les racines), on a*

$$\begin{aligned}
p(z) &= \frac{1}{m}\sum_{j=1}^{m} \prod_{k=1, k\neq j}^{m} (1 - w_k z) \\
&= \frac{1}{m}\sum_{j=1}^{m} \frac{1 - z^m}{1 - w_j z}.
\end{aligned}$$

   Or $p(z) = p(w_1 z) = \cdots = p_(w_m z)$ pour tout z. comme p est de degré au plus $m - 1$, cela implique que $p(z) = p(0) = 1$.
3. *On a donc $I - A^m = \prod_{k=1}^{m}(I - w_k A)$ et $I = \frac{1}{m}\sum_{j=1}^{m}\prod_{k=1, k\neq j}^{m}(1 - w_k A)$.*

4. *Pour x de norme 1, on a*

$$
\begin{aligned}
1 - x^H A^m x &= (Ix)^H (I - A^m) x \\
&= \left(\frac{1}{m} \sum_{j=1}^{m} \prod_{k=1, k\neq j}^{m} (1 - w_k A) x \right)^H \prod_{k=1}^{m} (I - w_k A) x \\
&= \frac{1}{m} \sum_{j=1}^{m} z_j^H (1 - w_j A) z_j, \text{ avec } z_j = \prod_{k=1, k\neq j}^{m} (1 - w_k A) x \\
&= \frac{1}{m} \sum_{j=1, z_j \neq 0}^{m} \|z_j\|^2 \left(1 - w_j \left(\frac{z_j}{\|z_j\|} \right)^H A \left(\frac{z_j}{\|z_j\|} \right) \right)
\end{aligned}
$$

5. *En remplacant A par $e^{i\theta} A$, on obtient*

$$
1 - e^{im\theta} x^H A^m x = \frac{1}{m} \sum_{j=1, z_j \neq 0}^{m} \|z_j\|^2 \left(1 - e^{i\theta} w_j \left(\frac{z_j}{\|z_j\|} \right)^H A \left(\frac{z_j}{\|z_j\|} \right) \right).
$$

Si $r(A) \leq 1$, la partie réelle du membre droit de cette égalité est positive ou nulle. En effet, $\mathrm{Re}(1 - e^{i\theta} w_j \left(\frac{z_j}{\|z_j\|} \right)^H A \left(\frac{z_j}{\|z_j\|} \right) |) \geq 1 - | \left(\frac{z_j}{\|z_j\|} \right)^H A \left(\frac{z_j}{\|z_j\|} \right) | \geq 0$, *ce qui implique que* $\mathrm{Re}(1 - e^{im\theta} x^H A^m x)$ *est positif ou nul. En prenant θ tel que $e^{im\theta} x^H A^m x = |x^H A^m x|$, on obtient $|x^H A^m x| \leq 1$, et donc $r(A^m) \leq 1$.*

□

Proposition 5.11 *Pour toute matrice carrée,*

$$
\frac{1}{2} \|A\| \leq r(A) \leq \|A\|.
$$

Preuve 5.11 *Démonstration. La partie droite est s'obtient en utilisant la sous-multiplicativité des normes. Pour la partie gauche, $A = \frac{1}{2}(A + A^H) + \frac{1}{2}(A - A^H)$ de sorte que*

$$
\|A\| \leq \frac{1}{2} (\|A + A^H\| + \|A - A^H\|).
$$

Comme $A \pm A^H$ est Hermitienne,

$$
\begin{aligned}
\|A \pm A^H\| &= \max_{z \neq 0} \frac{|z^H A z \pm z^H A^H z|}{z^H z} \\
&\leq r(A) + r(A^H) = 2 r(A).
\end{aligned}
$$

□

Proposition 5.12 *On a l'inégalité suivante*

$$
\|A x_j - b\| \leq 2 \left(\frac{r}{|c|} \right)^j \|A x_0 - b\|.
$$

Preuve 5.12 *Démonstration de la proposition. Soit Q_0 le polynôme donné par $Q_0(t) = \left(1 - \frac{t}{c}\right)^m$. Alors en utilisant les deux lemmes,$\|Q_0(A)\| \leq 2r(Q(A)) \leq 2r(I - A/c)^m$. Or $r(I - A/c) \leq r/|c|$. Donc*

$$\begin{aligned} \|Ax_j - b\| &= \min_{Q \in \mathcal{P}_j, Q(0)=1} \|Q(A)b\| \\ &\leq \|Q_0(A)b\| \leq \|Q_0(A)\|\|b\| \leq 2\left(\frac{r}{|c|}\right)^j \|Ax_0 - b\| \end{aligned}$$

□

5.3.3 Utilisation pratique de GMRES

5.3.4 Arrêt des itérations

Le critère d'arrêt présenté dans l'algorithme jusqu'ici consiste à détecter l'espace invariant $\mathcal{K}(A, b, j_s)$ en observant si $h_{j_s+1,j_s} = 0$. Ce type de test n'est jamais utilisé en pratique *car il est trop dangereux en présence d'erreur d'arrondis. On préfère s'arrêter lorsque les résidus normalisés (erreurs inverses) $\frac{\|Ax_k - b\|_2}{\|A\|_2\|x_k\|_2 + \|b\|_2}$ ou $\frac{\|Ax_k - b\|_2}{\|b\|_2}$ sont suffisamment petits. Il faut noter aussi que le calcul de $\|Ax_k - b\|_2$ pour le critère d'arrêt peut se faire implicitement lors de la résolution du problème de moindres carrés $\min \left\|\|b\|e_1 - \bar{H}_j z_j\right\|_2$, et ne nécessite pas de produit additionnel par A.*

De plus, même en présence d'erreurs d'arrondis, il a été démontré que la méthode non redémarrée décrite ci-dessus, appelée MGS GMRES permet d'obtenir une valeur de $\frac{\|Ax_k - b\|_2}{\|A\|_2\|x_k\|_2 + \|b\|_2}$ de l'ordre de la précision machine en n pas au plus (la méthode est dite inverse stable).

5.3.5 Préconditionnement

Les propositions ci-dessus permettent de donner des conditions suffisantes de réduction de la norme du résidu au cours d'un restart et donc d'obtenir des conditions de convergence de l'algorithme redémarré. Des techniques de transformations du système linéaire $Ax = b$ en un système équivalent pour lequel GMRES converge plus vite sont appelées techniques de préconditionnement. Les caractéristiques principales d'une bonne technique de préconditionnement sont :

- *ne pas être très coûteuse en place mémoire,*
- *sa mise en oeuvre (préparation + utilisation dans la méthode) ne doit pas engendrer trop de calculs,*
- *elle doit accélérer la méthode itérative.*

Pour les méthodes pour matrices non-symétriques comme GMRES, on parle fréquemment de préconditionnement

- *à gauche ; $Ax = b$ est remplacé par $M^{-1}Ax = M^{-1}b$ où M est inversible.*
- *à droite ; $Ax = b$ est remplacé par $AM^{-1}t = b$ et $x = M^{-1}t$, où M est inversible.*
- *mixte ; $Ax = b$ est remplacé par $M_1^{-1}AM_2^{-1}t = M_1b$ et $x = M_2^{-1}t$, où M_1 et M_2 sont inversibles. est inversible.*

Coût de la méthode (les termes en $O(k)$, $O(k^2)$,.., sont négligés)

- *Mémoire : stockage de A, de M_i et pour un vecteur de taille n* supplémentaire *à chaque pas*
- *Opérations : pour chaque étape, une application de A et inversion d'un système avec M_i, et $4kn$ opérations flottantes par itération.*

5.4 La méthode du gradient conjugué

Dans cette section, la matrice A est supposée symétrique définie positive. Soit $x^* = A^{-1}b$. La condition $b - Ax_k \perp \mathcal{K}(A, b, k)$ s'écrit

$$V_k^T(b - Ax_k) = 0.$$

Partant de $b = r_0 = \|r_0\|v_1$ (on suppose sans perdre de généralité que $x_0 = 0$) on a $V_k^T b = \|r_0\|e_1$. Comme de plus $x_k \in \mathcal{K}(A, b, k)$, $x_k = V_k y$ on obtient

$$V_k^T A V_k y_k = \|r_0\|e_1. \tag{5.2}$$

- *La matrice $V_k^T A V_k = H_k$ est générée par l'algorithme.*
- *Puisque A est symétrique, H_k est tridiagonale T_k.*
- *La matrice T_k est non singulière. En effet, si $T_k y = 0$ alors $V_k^T A V_k y = 0$ donc $y^T V_k^T A V_k y = 0$, ce qui implique $V_k y = 0$ car A est définie positive, et donc $V_k^T V_k y = y = 0$.*
- *L'itéré de Ritz-Galerkin est donc défini par $x_k = V_k(T_k^{-1}\|r_0\|e_1)$.*

Proposition 5.13 *Comme A est symétrique définie positive, la fonction $x \mapsto \sqrt{x^T A x}$ est une norme. La condition de Ritz-Galerkin devient $b - Ax_k \perp \mathcal{K}(A, b, k)$, d'où $A(x_k - x^*) \perp \mathcal{K}(A, b, k)$ ou encore, $(x_k - x^*) \perp_A \mathcal{K}(A, b, k)$. Cela signifie que x_k est tel que $\|x_k - x^*\|_A$ est minimum sur $\mathcal{K}(A, b, k)$.*

Preuve 5.13 *<u>Démonstration.</u> Soit $x_k = V_k y_k$, et $x = V_k y \in \mathcal{K}(A, b, k)$. Alors $\|x - x^*\|_A^2 = \|x - x_k + x_k - x^*\|_A^2 = \|x - x_k\|_A^2 + \|x_k - x^*\|_A^2 + 2(x_k - x)^T A(x_k - x^*)$. Comme $(x_k - x^*) \perp_A \mathcal{K}(A, b, k)$ et x_k et x sont tous deux dans $\mathcal{K}(A, b, k)$, on a $\|x - x^*\|_A^2 = \|x - x_k\|_A^2 + \|x_k - x^*\|_A^2 \geq \|x_k - x^*\|_A^2$.*

□

Proposition 5.14 *Si la méthode s'arrête ($AV_k = V_k T_k$), alors x_k est solution du problème.*

Preuve 5.14 *<u>Démonstration.</u> On a en effet $AV_k y_k - r_0 = V_k(T_k y_k - \|r_0\|e_1) = 0$.*

□

Proposition 5.15 *La méthode RGM converge en au plus m itérations sur une matrice ayant m valeurs propres distinctes.*

Preuve 5.15 *Démonstration. Suposons que A a m valeurs propres distinctes λ_i, $i = 1, \ldots m$, et soit $p(\lambda) = \prod_{i=1}^{m}(\lambda - \lambda_i)$. Alors en diagonalisant $A = QDQ^T$, on obtient $p(A) = Qp(D)Q^T = 0$. En reprenant la démonstration de la proposition 5.1, on obtient que la solution appartient à l'espace de Krylov de d'ordre m.*

□

5.4.1 Convergence de la méthode de Ritz-Galerkin (RGM)

Comme $x_k \in x_0 + \mathcal{K}(A, b, k)$, on a $x_k = x_0 + Q_{k-1}(A)r_0$ où Q_{k-1} est un polynôme de degré au plus $k-1$. On a alors

$$\begin{aligned} x_k - x^* &= x_0 + Q_{k-1}(A)(Ax^* - Ax_0) - x^* && (5.3)\\ &= (I - Q_{k-1}(A)A)(x_0 - x^*) && (5.4)\\ &= (I - AQ_{k-1}(A))(x_0 - x^*), && (5.5) \end{aligned}$$

ce qui montre que

$$\|x_k - x^*\|_A = \|(I - AQ_{k-1}(A))(x_0 - x^*)\|_A.$$

La minimalité de $\|x_k - x^*\|_A$ sur l'espace $\mathcal{K}(A, b, k)$ entraîne la proposition suivante.

Proposition 5.16 *Le polynôme $Q_{k-1}(A)$ construit par la procédure RGM vérifie*

$$\|(I - AQ_{k-1}(A))(x_0 - x^*)\|_A = \min_{Q \in P_{k-1}} \|(I - AQ(A))(x_0 - x^*)\|_A,$$

où P_{k-1} est l'ensemble des polynômes de degré au plus $k-1$.

Proposition 5.17 *Soit x_k le kième itéré de la RGM. On a*

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

Preuve 5.16 *Démonstration La Proposition 5.16 permet d'écrire*

$$\|x_k - x^*\|_A = \min_{p \in P_k, p(0)=1} \|p(A)(x_0 - x^*)\|_A.$$

Soient λ_i, $i = 1, ..., n$ les valeurs propres de A et $\xi = \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix}$ où ξ_i $i = 1, ..., n$ sont les composantes de $(x_0 - x^*)$ dans la base constituée des colonnes de V. On a $A = V\Lambda V^T$ et $(x_k - x^*) = V\xi$ ce qui entrîne

$$\begin{aligned} p(A)(x_0 - x^*) &= Vp(\Lambda)V^T(V\xi)\\ &= Vp(\Lambda)\xi. \end{aligned}$$

$$\begin{aligned}
\|p(A)(x_0 - x^*)\|_A^2 &= (Vp(\Lambda)\xi)^T A (Vp(\Lambda)\xi) \\
&= \sum_{i=1}^{n} p(\lambda_i)^2 \lambda_i \xi_i^2 \\
&\le \max_i (p(\lambda_i)^2) \sum_{i=1}^{n} \lambda_i \xi_i^2 \\
&\le \max_i (p(\lambda_i)^2) \|x_0 - x^*\|_A^2 \\
&\le \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} (p(\lambda))^2 \|x_o - x^*\|_A^2 .
\end{aligned}$$

Ceci montre que

$$\|x_k - x^*\|_A \le \min_{p \in P_k, p(0)=1} \left(\max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |p(\lambda)| \right) \|x_o - x^*\|_A . \tag{5.6}$$

Un résultat d'approximation par les polynômes de Chebyshev montre que

$$\min_{p \in P_k, p(0)=1} \left(\max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |p(\lambda)| \right) \le \frac{1}{|C_m(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}})|} \tag{5.7}$$

où $C_k(t)$ est un polynôme de Chebyshev de première espèce et de degré k. Pour $|t| > 1$ on a

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^{-k} \right] \ge \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k .$$

En posant $\eta = \frac{\lambda_{min}}{\lambda_{\max}-\lambda_{\min}}$ on a $\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}} = 1 + 2\eta$

$$\begin{aligned}
C_k\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right) &= C_k(1+2\eta) \\
&\ge \frac{1}{2}\left(1 + 2\eta + \sqrt{(1+2\eta)^2 - 1}\right)^k \\
&\ge \frac{1}{2}\left(1 + 2\eta + 2\sqrt{\eta(\eta+1)}\right)^k \\
&\ge \frac{1}{2}\left(\left(\sqrt{\eta} + \sqrt{\eta+1}\right)^2\right)^k \\
&\ge \frac{1}{2}\left(\frac{(\sqrt{\lambda_{min}} + \sqrt{\lambda_{max}})^2}{\lambda_{\max} - \lambda_{\min}}\right)^k \\
&\ge \frac{1}{2}\left(\frac{\sqrt{\lambda_{max}} + \sqrt{\lambda_{min}}}{\sqrt{\lambda_{\max}} - \sqrt{\lambda_{\min}}}\right)^k \\
&\ge \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k .
\end{aligned}$$

Cela implique

$$\frac{1}{|C_m(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}})|} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k .$$

ce qui permet de compléter la preuve en utilisant (5.7) et (5.6).

□

En utilisant les propositions 5.17 et 5.15, il apparaît qu'une technique visant a remplacer le système d'origine $Ax = b$ en un système équivalent

- *mieux conditionné, ou bien où,*
- *les valeurs propres distinctes sont moins nombreuses,*

permet d'accélerer la convergence de la méthode. Plus généralement, on appelle préconditionnement toute technique visant a accélérer (en temps de calcul, ou en nombre d'itération) une méthode itérative.

On rappelle les caractéristiques principales d'une bonne technique de préconditionnement sont :

- *ne pas être très coûteuse en place mémoire ,*
- *sa mise en oeuvre (préparation + utilisation dans la méthode) ne doit pas engendrer trop de calculs,*
- *elle doit accélérer la méthode itérative.*

5.4.2 La méthode du gradient conjugué en pratique

Forme classique

La RCM permet de définir de manière unique une suite d'itérés. Cette méthode peut être implantée de différentes manières dans les logiciels de calculs. La méthode la plus stable en présence d'erreurs d'arrondis est la méthode du gradient conjugué. Nous donnons ici l'algorithme sous sa forme la plus stable. Cette forme est dérivée dans de nombreux ouvrages tels que "Matrix Computations " de Golub et Van Loan.

Conjugate Gradient algorithm (CG)	
1.	Compute $r_0 = b - Ax_0$ and $p_0 = r_0$
2.	For k=0,2, ... Do
3.	$\quad \alpha_k = r_k^T r_k / p_k^T A p_k$
4.	$\quad x_{k+1} = x_k + \alpha_k p_k$
5.	$\quad r_{k+1} = r_k - \alpha_k A p_k$
6.	$\quad \beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$
7.	$\quad p_{k+1} = r_{k+1} + \beta_k p_k$
8.	if converged then stop
9.	EndDo

Le critère d'arrêt prend en pratique la forme de résidus normalisés cités ci-dessus : $\frac{\|Ax_k - b\|_2}{\|A\|_2 \|x_k\|_2 + \|b\|_2}$ *ou* $\frac{\|Ax_k - b\|_2}{\|b\|_2}$.

Préconditionnement

Contrairement aux méthodes pour matrices nonsymétriques, le préconditionnement de CG doit toujours garantir que la matrice préconditionnée est symétrique définie positive. Pour cela on impose que le préconditionneur M^{-1} est symétrique défini positif. Dans ce cas, une factorisation de Cholesky donne $M^{-1} = CC^T$. Une idée naturelle est de remplacer *le système d'origine par le système* $C^TAC\tilde{x} = C^Tb$. *On pose* $\tilde{A} = C^TAC$, $C\tilde{x} = x$, $\tilde{b} = C^Tb$ *et*

$$\begin{aligned} x_k &= C\tilde{x}_k, \\ C\tilde{p}_k &= p_k, \\ \tilde{r}_k &= C^T r_k, \\ z_k &= CC^T r_k. \end{aligned}$$

L'algorithme s'écrit de deux manières équivalentes :

Conjugate Gradient algorithm		
1.	*Compute* $\tilde{r}_0 = \tilde{b} - \tilde{A}\tilde{x}_0$ *and* $\tilde{p}_0 = \tilde{r}_0$	
2.	*For k=0,2, ... Do*	
3.	$\alpha_k = \tilde{r}_k^T\tilde{r}_k/\tilde{p}_k^T\tilde{A}\tilde{p}_k$	$= r_k^TCC^Tr_k/p_k^TAp_k = r_k^Tz_k/p_k^TAp_k$
4.	$\tilde{x}_{k+1} = \tilde{x}_k + \alpha_k\tilde{p}_k$	$\overset{C}{\Rightarrow} x_{k+1} = x_k + \alpha_k p_k$
5.	$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k\tilde{A}\tilde{p}_k$	$\overset{C^T}{\Rightarrow} r_{k+1} = r_k - \alpha_k Ap_k$
6.	$\beta_k = \tilde{r}_{k+1}^T\tilde{r}_{k+1}/\tilde{r}_k^T\tilde{r}_k$	$= r_{k+1}^TCC^Tr_{k+1}/r_k^TCC^Tr_k = r_{k+1}^Tz_{k+1}/r_k^Tz_k$
7.	$\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k\tilde{p}_k$	$\overset{C}{\Rightarrow} p_{k+1} = r_{k+1} + \beta_k p_k$
8.	*if converged then stop*	
9.	*EndDo*	

Cela nous donne finalement l'algorithme du gradient conjugué préconditionné, où l'on voit que l'on n'a plus besoin du facteur de Cholesky de M^{-1}, *mais simplement de* résolution de systèmes linéaires *avec* M.

Preconditioned Conjugate Gradient algorithm	
1.	*Compute* $r_0 = b - Ax_0$, $z_0 = M^{-1}r_0$ *and* $p_0 = r_0$
2.	*For k=0,2, ... Do*
3.	$\alpha_k = r_k^Tr_k/p_k^TAp_k$
4.	$x_{k+1} = x_k + \alpha_k p_k$
5.	$r_{k+1} = r_k - \alpha_k Ap_k$
6.	$z_{k+1} = M^{-1}r_{k+1}$
7.	$\beta_k = r_{k+1}^Tr_{k+1}/r_k^Tr_k$
8.	$p_{k+1} = r_{k+1} + \beta_k p_k$
9.	*if converged then stop*
10.	*EndDo*

Il est possible de montrer que, dans cet algorithme, les résidus r_k *sont* $M^{-1}-$*orthogonaux* ($r_k^TM^{-1}r_l = \delta_{kl}$) *et que les* p_k *sont* $A-$*orthogonaux* ($p_k^TAp_l = \delta_{kl}$). *Les* p_k *sont appelés aussi directions de descente d'après une interprétation en terme d'algorithme d'optimisation. Coût de la méthode*

- *Mémoire : stockage de* A, *du préconditionneur* M *et de* 4 *vecteurs de taille* n ($A \in \mathbb{R}^{n\times n}$)
- *Opérations : pour chaque étape, une application de* A *et une résolution d'un système linéaire avec* M, *et* $10n$ *opérations flottantes par itération.*

Cependant, les erreurs d'arrondis dans la méthode font qu'en pratique la solution peut ne pas être obtenue en n *pas. Des techniques coûteuses de réorthogonalisation permettent de diminuer quelque peu l'impact de ces erreurs.*