Le rayon spectral est donc l'information déterminante pour étudier la convergence des suites de matrices.

Objectif : Construire des algorithmes itératifs de résolution d'un système linéaire et définir leurs propriétés et avantages.

1.4 Méthodes itératives

Dans la première partie de ce chapitre, nous nous sommes intéressés à des méthodes directes pour résoudre le problème linéaire. On rappelle les deux approches :

1. la factorisation LU et ses extensions (permutations, Cholesky, matrices bandes)
2. la factorisation QR

Les méthodes directes se caractérisent par le fait que la résolution s'effectue en un nombre fini d'étapes. Les méthodes directes sont très efficaces : elles donnent la solution exacte (aux erreurs d'arrondi près) du système linéaire considéré. Elles ont l'inconvénient de nécessiter une assez grande place mémoire car elles nécessitent le stockage de toute la matrice en mémoire vive.

Cependant, si le système a été obtenu à partir de la discrétisation d'équations aux dérivés partielles, il est en général "creux", c.à.d. qu'un grand nombre des coefficients de la matrice du système sont nuls ; de plus la matrice a souvent une structure "bande", i.e. les éléments non nuls de la matrice sont localisés sur certaines diagonales. On a vu que dans ce cas, la méthode de Cholesky "conserve le profil" (voir TD).

Lorsqu'on a affaire à de très gros systèmes issus de l'ingénierie (mécanique des fluides, calcul des structures, ...), où n peut être de l'ordre de plusieurs milliers, on cherche à utiliser des méthodes nécessitant le moins de mémoire possible.

Dans cette partie on s'intéresse à des méthodes itératives qui, à la différence des méthodes directes, convergent théoriquement en un nombre infini d'étapes. "L'espoir" est qu'en peu d'itérations, la solution approchée soit très proche de la solution exacte et que le coût des itérations soit faible.

Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ une matrice inversible et $\mathbf{b} \in \mathbb{R}^n$, on cherche toujours ici à résoudre le système linéaire $A\mathbf{x} = \mathbf{b}$, mais de façon itérative, c.à.d. par la construction d'une suite.

Définition 1.61. On appelle méthode itérative de résolution du système linéaire $A\mathbf{x} = \mathbf{b}$ une méthode qui construit une suite $(\mathbf{x}^k)_{k\in\mathbb{N}}$ (où l'itéré $\mathbf{x}^k$ est calculé à partir des itérés précédent $\mathbf{x}^0, \ldots, \mathbf{x}^{k-1}$) convergente vers $\mathbf{x}$ solution du système.

Définition 1.62. On dit qu'une méthode itérative est convergente si pour tout choix initial $\mathbf{x}^0 \in \mathbb{R}^n$, on a :

Pour résoudre le système linéaire $A \in \mathbb{M}_{n\times n}(\mathbb{R})$, $\mathbf{b} \in \mathbb{R}^n$:

$$A\mathbf{x} = \mathbf{b},$$

une idée naturelle est de travailler avec une matrice M inversible qui soit "proche" de A, mais plus facile que A à inverser. On appelle cette matrice M matrice de préconditionnement. On écrit alors $A = M - N$ et on réécrit le système linéaire comme suit $M\mathbf{x} = (M - A)\mathbf{x} + \mathbf{b} = N\mathbf{x} + \mathbf{b}$.

Cette forme suggère la construction de la suite $\mathbf{x}^{k+1}$

(4)

Remarque. Si l'algorithme converge, i.e. $\mathbf{x}^k \to \mathbf{x}^*$, alors il converge vers la solution du problème initial.

L'objectif sera de choisir convenablement M pour que le calcul de $M^{-1}\mathbf{y}$ soit peu coûteux, pour que l'algorithme converge et qu'il converge vite. On rappelle (définition 1.62) que la méthode itérative (4) converge si $\forall\, \mathbf{x}_0 \in \mathbb{R}^n \lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}$

Proposition 1.63. *La méthode converge si et seulement si* $\rho(M^{-1}N) < 1$.

Démonstration.

□

Critère d'arrêt : En théorie, il faudrait effectuer un nombre infini d'itérations pour obtenir la solution exacte d'un système linéaire avec une méthode itérative. En pratique, ce n'est ni nécessaire, ni raisonnable. L'objectif étant de faire peu d'itérations, il important d'avoir un bon critère d'arrêt de la méthode :

- Un premier estimateur est donné par le résidu $\mathbf{r}^k = \mathbf{b} - A\mathbf{x}^k$ comme suit :

$$\|\mathbf{r}^k\| \leq \varepsilon \|\mathbf{b}\|,$$

avec ce critère on a

Ce test est donc pertinent si le conditionnement de A n'est pas trop grand.

- Un autre estimateur est donné par l'incrément

$$\delta^k = \mathbf{x}^{k+1} - \mathbf{x}^k.$$

On peut montrer que le contrôle par l'incrément n'est pertinent que quand $\rho(M^{-1}N) \ll 1$.

En résumé, on a deux critères d'arrêt possible pour les méthodes itératives : l'un basé sur le résidu, l'autre sur l'incrément. Le premier est pertinent quand le système est bien conditionné, le second quand le rayon spectral de la matrice d'itération n'est pas trop proche de 1.

Estimation de la vitesse de convergence : Soit $\mathbf{x}^0 \in \mathbb{R}^n$ donné et soit $(\mathbf{x}^k)_{k\geq 0}$ la suite définie par (4). Notons la matrice d'itération $B = M^{-1}N$. On a vu que, si $\rho(B) < 1$, alors $\mathbf{x}^k \to \mathbf{x}$ quand $k \to \infty$, où $\mathbf{x}$ est la solution du système $A\mathbf{x} = \mathbf{b}$. On peut montrer (sauf cas particuliers qu'on ne précise pas ici)

Proposition 1.64 (Admis)**.**

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}\|}{\|\mathbf{x}^k - \mathbf{x}\|} \to \rho(B) \text{ } \textit{lorsque } k \to +\infty.$$

indépendamment de la norme choisie sur $\mathbb{R}^n$*.*

Remarque.

1. Le rayon spectral $\rho(B)$ de la matrice B est donc une bonne estimation de la vitesse de convergence.
2. Pour estimer cette vitesse de convergence lorsqu'on ne connaît pas x, on peut utiliser le fait qu'on a aussi

$$\frac{\|\mathbf{x}^{k+1} - \mathbf{x}\|}{\|\mathbf{x}^k - \mathbf{x}\|} \to \rho(B) \text{ lorsque } k \to +\infty.$$

En résumé, l'étude des méthodes itératives repose donc sur deux questions suivantes :

- étant donné une méthode itérative de matrice $B = M^{-1}N$, déterminer si la méthode est convergente, c'est-à-dire si $\rho(M^{-1}N) < 1$, ou de façon équivalente, s'il existe une norme matricielle telle que $\|M^{-1}N\| < 1$.
- étant donné deux méthodes itératives convergentes, les comparer : la méthode itérative la plus "rapide" est celle dont la matrice a le plus petit rayon spectral.

Enfin, l'algorithme général s'écrit de la façon suivante :

Algorithme 7 : Algorithme itératif ($A = M - N$)

$\mathbf{x} = \mathbf{x}^0$
$\mathbf{r}^0 = A\mathbf{x} - \mathbf{b}$
tant que $\|\mathbf{r}\| \geq \varepsilon\|\mathbf{r}^0\|$ **faire**
  $\mathbf{y} = M^{-1}\mathbf{r}$
  $\mathbf{x} = \mathbf{x} - \mathbf{y}$
  $\mathbf{r} = A\mathbf{x} - \mathbf{b}$
fin

On a bien $\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{y}^k = \mathbf{x}^k - M^{-1}(M - N)\mathbf{x}^k + M^{-1}\mathbf{b} = M^{-1}N\mathbf{x}^k + M^{-1}\mathbf{b}$

Remarque. Pour la norme 2, on effectue le test d'arrêt directement sur le carré des normes pour éviter le calcul de la racine.

1.4.1 Méthode de Jacobi et de Gauss-Seidel

Soit $A = D - E - F$ la décomposition de A définie par :

$$D = (d_{ij})_{i,j=1,\dots,n} \quad \text{avec} \quad \begin{cases} d_{ii} = a_{ii} & i = 1, \dots, n \\ d_{ij} = 0 & i \neq j \end{cases}$$

$$E = (e_{ij})_{i,j=1,\dots,n} \quad \text{avec} \quad \begin{cases} e_{ij} = -a_{ij} & i > j \\ e_{ij} = 0 & i \leqslant j \end{cases}$$

$$F = (f_{ij})_{i,j=1,\dots,n} \quad \text{avec} \quad \begin{cases} f_{ij} = -a_{ij} & i < j \\ f_{ij} = 0 & i \geqslant j \end{cases}$$

Méthode de Jacobi **Méthode de Gauss-Seidel**

On pose :

$$\begin{cases} M = D \\ N = F + E \end{cases} \qquad \begin{cases} M = D - E \\ N = F \end{cases}$$

La matrice d'itérations s'écrit :

$$B = D^{-1}(E + F) \qquad B = (D - E)^{-1}F$$

Les composantes du vecteur $\mathbf{x}^{k+1} = (x_i^{k+1})_{i=1}^n$ sont alors

Remarque.

1. Les deux méthodes nécessite $A_{ii} \neq 0$.
2. Coût / itérations de la méthode de Jacobi est $\mathcal{O}(2n^2)$ et de la méthode de Gauss-Seidel est $\mathcal{O}(3n^2)$.

Comparaison :

Méthode de Jacobi :

Méthode de Gauss-Seidel :

Pour calculer une composante quelconque x_i^{k+1} par la méthode de Jacobi on utilise donc $(n-1)$ des composantes du vecteur $\mathbf{x}^k$, qu'il faut donc garder en mémoire

pendant tout le calcul du vecteur $\mathbf{x}_{k+1}$. Autrement dit, une itération de la méthode immobilise $2n$ mémoires de l'ordinateur. La méthode de Gauss-Seidel permet utiliser "mieux" les quantités déjà calculées en utilisant la "nouvelle" valeur x_i^{k+1} plutôt que l'"ancienne" valeur x_i^k. Par conséquent, on peut espérer que la méthode de G.S. converge plus vite que celle de Jacobi. En revanche, la méthode de Jacobi présente l'avantage d'être facilement parallélisable.

Définition 1.65. On dit qu'une matrice est à diagonale strictement dominante si et seulement si

Proposition 1.66. *Si la matrice A est à diagonale strictement dominante, alors les méthodes de Jacobi et Gauss-Seidel convergent.*

Démonstration.

□

1.4.2 Méthode de descente

Considérons le cas d'une matrice A symétrique définie positive.

Proposition 1.67. *Si $A = A^t > 0$ le produit $(A\mathbf{x}, \mathbf{y}) = \mathbf{y}^t A\mathbf{x}$ définit un produit scalaire et on lui associe la norme : $\|\mathbf{x}\|_A = \sqrt{(A\mathbf{x}, \mathbf{x})}$.*

Theorem 1.68. *Soit $\mathbf{x}^*$ la solution de $A\mathbf{x} = \mathbf{b}$. Alors, $\mathbf{x}^*$ réalise le*

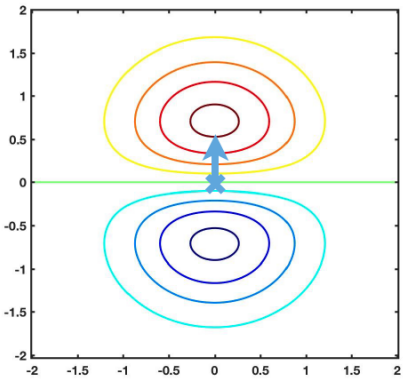
Démonstration.

□

Donc pour résoudre le problème linéaire on doit définir $\mathbf{x}^*$ minimisant $\mathcal{J}(\mathbf{x})$: en partant du point initial $\mathbf{x}^0 \in \mathbb{R}^n$ on doit choisir une direction appropriée pour s'approcher de la solution $\mathbf{x}^*$. Le choix optimale de la direction est inconnu à priori, mais on sait que le gradient d'une fonction est dirigé vers la direction de plus fort accroissement, donc une idée naturelle de choisir cette direction. Or, $\nabla\mathcal{J}(\mathbf{x}^k) = A\mathbf{x}^k - \mathbf{b}$, et donc la direction du gradient coïncide avec celle du résidu et on peut la calculer à partir de l'itérée $\mathbf{x}^k$. D'où l'algorithme de descente :

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha \underbrace{\left(A\mathbf{x}^k - \mathbf{b}\right)}_{=\nabla\mathcal{J}(\mathbf{x}^k)} \tag{5}$$

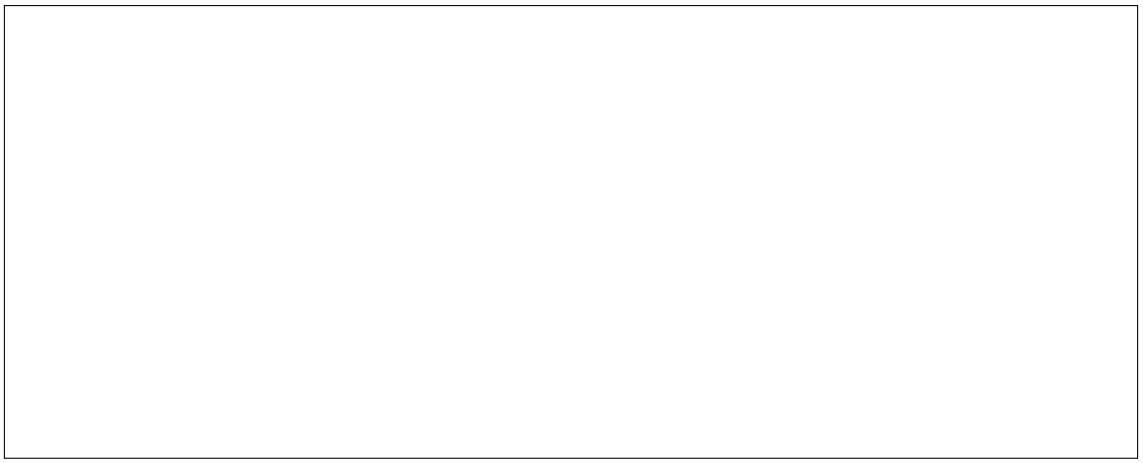
le choix de α (appelé le pas de descente) est très important. On peut réécrire l'itération sous la forme : $\mathbf{x}^{k+1} = \underbrace{(I - \alpha A)}_{M^{-1}N}\mathbf{x}^k + \alpha\mathbf{b}$.



Theorem 1.69. *L'algorithme* (5) *est convergent si et seulement si le pas vérifie* $0 < \alpha < \frac{2}{\lambda_n}$.

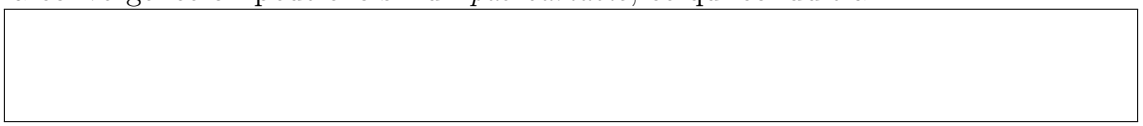
Démonstration.

□



□

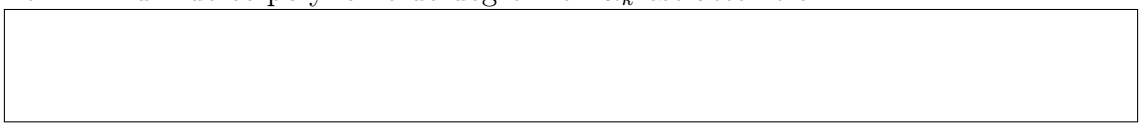
On constate que le choix de α a une influence sur la convergence de la méthode de descente. Le résultat du théorème précédent est bien théoriquement, mais en pratique on ne pourra pas s'en servir. Effectivement, il faudrait connaître la valeur propre maximal λ_n, ce qui n'est pas possible en temps raisonnable. Pour améliorer la convergence on peut choisir un *pas variable*, ce qui conduit à



On va alors choisir le pas a chaque étape afin de minimiser :

$$\begin{aligned}\mathcal{J}\left(\mathbf{x}^{k+1}\right) &= \frac{1}{2}\left(A\mathbf{x}^k - \alpha_k A\mathbf{r}^k, \mathbf{x}^k - \alpha_k \mathbf{r}^k\right) - \left(b, \mathbf{x}^k - \alpha_k \mathbf{r}^k\right) \\ &= \frac{1}{2}\alpha_k^2\left(A\mathbf{r}^k, \mathbf{r}^k\right) - \alpha_k \left\|\mathbf{r}^k\right\|_2^2 + \text{const.}\end{aligned}$$

Le minimum de ce polynôme de degré 2 en α_k est atteint en



En résumé, on a les deux algorithmes suivants :

Algorithme 7 : Pas fixe

$\mathbf{x} = \mathbf{x}^0$
$\mathbf{r}^0 = A\mathbf{x} - \mathbf{b}$
tant que $\|\mathbf{r}\| \geq \varepsilon\|\mathbf{r}^0\|$ **faire**
  $\mathbf{x} = \mathbf{x} - \alpha\mathbf{r}$
  $\mathbf{r} = A\mathbf{x} - \mathbf{b}$
fin

Algorithme 8 : Pas variable

$\mathbf{x} = \mathbf{x}^0$
$\mathbf{r}^0 = A\mathbf{x} - \mathbf{b}$
tant que $\|\mathbf{r}\| \geq \varepsilon\|\mathbf{r}^0\|$ **faire**
  $\alpha = (\mathbf{r}, \mathbf{r})/(A\mathbf{r}, \mathbf{r})$
  $\mathbf{x} = \mathbf{x} - \alpha\mathbf{r}$
  $\mathbf{r} = A\mathbf{x} - \mathbf{b}$
fin

En conclusion :

- Résoudre un système linéaire avec une méthode itérative consiste à construire, en partant d'une donnée initiale $\mathbf{x}^0$, une suite de vecteurs $\mathbf{x}^k$ convergeant vers la solution exacte quand $k \to \infty$;
- une méthode itérative converge si pour toute donnée initiale $\mathbf{x}^0$ on a $\mathbf{x}^k \to \mathbf{x}^*$ quand $k \to \infty$;

- une méthode itérative converge si et seulement si le rayon spectral de la matrice d'itération ($B = M^{-1}N$) est strictement plus petit que 1 ;
- les méthodes itératives traditionnelles sont celles de Jacobi et de Gauss-Seidel. Une condition suffisante de convergence est que la matrice soit à diagonale strictement dominante par ligne ;
- dans la méthode de descente, la convergence est accélérée à l'aide d'un paramètre ;

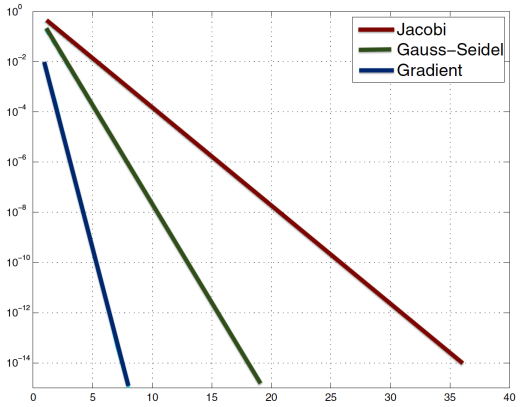


FIGURE 11 – Comparaison de convergence des méthodes (A. Quarteroni, R. Sacco, F. Saleri "Numerical Mathematics")

1.4.3 Les méthodes de Krylov

La méthode du gradient reprend souvent la même direction, on essaie donc de l'améliorer en forçant une direction différente à chaque itération.

Définition 1.70. Soit $\mathbf{x}_0$, on note $\mathbf{r}_0 = A\mathbf{x}_0 - b$. L'espace de Krylov d'ordre k associé est défini par

Si on considère la méthode de descente avec pas variable, on peut voir que $\mathbf{r}^k = \prod_{j=0}^{k-1}(I - \alpha_j A)\,\mathbf{r}^0$ et donc $\mathbf{r}^k = p_k(A)\mathbf{r}^0$ ou $p_k(A)$ est un polynôme de degré k, et on remarque immédiatement que $\mathbf{r}^k \in K_{k+1}(A, \mathbf{r}_0)$.

De façon similaire, l'itérée de la méthode de descente avec pas variable étant donné par $\mathbf{x}^k = \mathbf{x}^0 + \sum_{j=0}^{k-1} \alpha_j \mathbf{r}^j$, on a donc

Donc de manière plus général on peut réécrire la méthode de descente comme suit

$$\mathbf{x}^k = \mathbf{x}^0 + q_{k-1}(A)\mathbf{r}^0.$$

ou $q_{k-1}(A)$ est un polynôme en A. Ainsi on cherche la solution non pas directement dans l'espace $\mathbb{R}^n$ mais dans le sous-espace W_k. Le résultat suivant nous permet de dire que cette approche semble être raisonnable :

Proposition 1.71. *Soit $A \in \mathbb{M}_{n\times n}(\mathbb{R})$ et $\mathbf{v} \in \mathbb{R}^n$. La dimension de $K_k(A;\mathbf{v})$ est égal à k si et seulement si le degré $\deg_A v \geq k$. Ou $\deg_A \mathbf{v}$ est défini par*

$$\deg_A \mathbf{v} := \min\{\deg(P(A)) \text{ où } P(A)\mathbf{v} = \mathbf{0}\}$$

Et donc par théorème de Cayley-Hamilton $\deg_A \mathbf{v} \leq n$, il est possible qu'on cherche la solution dans l'espace de Krylov de dimension petite devant la dimension n.

Deux stratégie sont possible :

- chercher $\mathbf{x}^k \in W_k$ tel que $\mathbf{r}^k$ est orthogonal à $K_k(A;\mathbf{r}^0)$:
$$\mathbf{v}^T\left(\mathbf{b} - A\mathbf{x}^{(k)}\right) = 0 \quad \forall \mathbf{v} \in K_k\left(A;\mathbf{r}^{(0)}\right)$$
- chercher $\mathbf{x}^k \in W_k$ minimisant la norme de résidu $\|\mathbf{r}^k\|_2$
$$\left\|\mathbf{b} - A\mathbf{x}^{(k)}\right\|_2 = \min_{\mathbf{v}\in W_k} \|\mathbf{b} - A\mathbf{v}\|_2$$

Proposition 1.72. *Il existe un entier $k_0 \leq n$ critique tel que*

$$K_0(A;\mathbf{r}_0) \subsetneq K_1(A;\mathbf{r}_0) \subsetneq \ldots \subsetneq K_{k_0}(A;\mathbf{r}_0) = K_{k_0+1}(A;\mathbf{r}_0) = \ldots = K_n(A;\mathbf{r}_0)$$

Démonstration.

□

De plus, on peut montrer que k_0 satisfait la propriété suivante (voir preuve de la prop. suivante) :

$$k_0 = \min\{k \mid A^{-1}\mathbf{r}_0 \in K_n(A, \mathbf{r}_0)\}.$$

La proposition suivante établit le résultat général pour la solution exacte de $A\mathbf{x} = \mathbf{b}$:

Proposition 1.73. *La solution* $\mathbf{x} = A^{-1}\mathbf{b}$ *appartient à l'espace* $W_{k_0} = x^0 + K_{k_0}(A; \mathbf{r}_0)$

Démonstration.

□

1.4.4 Méthode de gradient conjugué

La méthode du gradient conjugué s'applique dans le cas où la matrice A est **symétrique définie positive**.

Soit $A = A^t > 0$. On définit $(\mathbf{x}^k)$

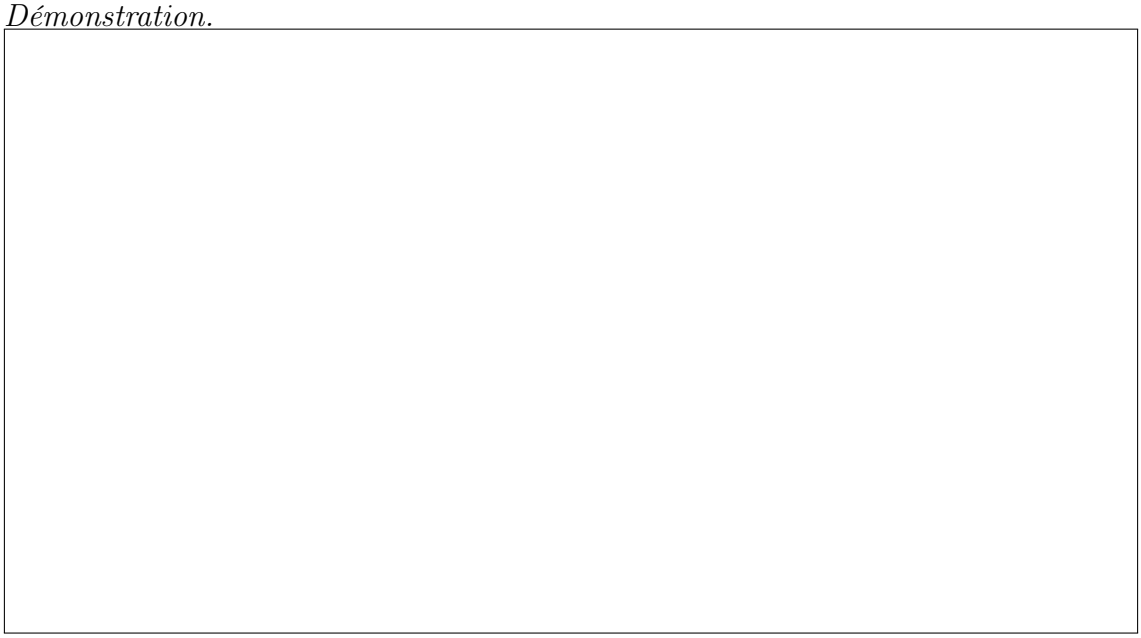
$$\begin{cases} \mathbf{x}^k \in W_k = \mathbf{x}^0 + K_k(A; \mathbf{r}_0) \\ \mathbf{v}^t \left(A\mathbf{x}^k - \mathbf{b}\right) = 0 \quad \forall \mathbf{v} \in K_k(A; \mathbf{r}_0) \end{cases} \tag{6}$$

Proposition 1.74. *Soit* k_0 *l'entier critique de la proposition 1.72 pour le quel* $K_{k_0}(A; \mathbf{r}_0) = K_{k_0+1}(A; \mathbf{r}_0)$. *Alors on a* $A\mathbf{x}^{k_0} = \mathbf{b}$ *et donc* $\mathbf{x}^{k_0}$ *est la solution exact.*

Version pratique du gradient conjugué : On considère une matrice symétrique définie positive $A \in \mathbb{M}_{n\times n}$ et $\mathbf{x}^0$, $\mathbf{b} \in \mathbb{R}^n$ fixés, et on pose $\mathbf{r}^0 := \mathbf{p}^0 := \mathbf{b} - A\mathbf{x}^0$. On définit alors les trois suites $(\mathbf{x}^k)_{k\geq 0}$, $(\mathbf{r}^k)_{k\geq 0}$, et $(\mathbf{p}^k)_{k\geq 0}$ par :

Alors la suite des $\mathbf{x}^k$ coïncide avec la suite des itérés de la méthode du gradient conjugué (6).

Démonstration.



Nous venons d'établir que la suite des $\mathbf{x}^k$ construite vérifie la première équation de (6). Pour conclure, il reste donc à démontrer qu'on a

$$\forall\, \mathbf{v} \in K_k, \quad \big(A\mathbf{x}^k - \mathbf{b}, \mathbf{v}\big) = 0 \quad \Leftrightarrow (\mathbf{r}^k, \mathbf{v}) = 0.$$

Pour ceci nous allons démontrer par récurrence sur $k \geq 0$ qu'on a

$$\forall\, j = 0 \ldots k-1, \quad \left\{ \begin{array}{rl} \big(\mathbf{r}^k, \mathbf{r}^j\big) & = 0 \\ \big(A\mathbf{p}^k, \mathbf{p}^j\big) & = 0 \end{array} \right. . \tag{7}$$

Il n'y a rien à démontrer pour $k = 0$. Supposons donc que ce soit vrai pour k, et démontrons que c'est encore vrai pour $k+1$. En utilisant la définition de $\mathbf{r}^k$ et de $\mathbf{p}^k$, on trouve

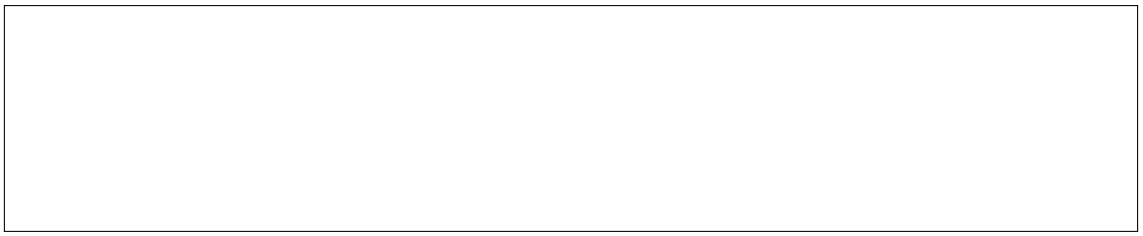
D'après l'hypothèse de récurrence on a :

$$\forall\, j = 0 \ldots k-1, \quad (\mathbf{r}^k, \mathbf{r}^j) = (A\mathbf{p}^k, \mathbf{p}^j) = (A\mathbf{p}^k, \mathbf{p}^{j-1}) = 0.$$

On vient donc de montrer que $(\mathbf{r}^{k+1}, \mathbf{r}^j) = 0$ pour $j < k$. Dans le cas $j = k$, on trouve (par le calcul ci-dessus) que $(\mathbf{r}^{k+1}, \mathbf{r}^k) = \|\mathbf{r}^k\|_2^2 - \alpha_k \|\mathbf{p}\|_A^2$, et alors l'expression de α_k nous permet de conclure que $(\mathbf{r}^{k+1}, \mathbf{r}^k) = 0$. En conclusion on a établi que

$$\forall\, j = 0 \ldots k, \quad (\mathbf{r}^{k+1}, \mathbf{r}^j) = 0. \tag{8}$$

Démontrons maintenant qu'on a également $(A\mathbf{p}^{k+1}, \mathbf{p}^j) = 0$ pour $j = 0 \ldots k$. Par definition de $\mathbf{p}^k$, et puisque $A\mathbf{p}^j = (\mathbf{r}^j - \mathbf{r}^{j+1})/\alpha_j$ (par definition de $\mathbf{r}^k$) on obtient



Dans la dernière égalité, le premier terme est nul puisque dans tous les cas on a $(\mathbf{r}^{k+1}, \mathbf{r}^j) = 0$. Par ailleurs si $j < k$, on a d'une part $(\mathbf{r}^{k+1}, \mathbf{r}^{j+1}) = 0$ d'après (8), et d'autre part $(A\mathbf{p}^k, \mathbf{p}^j) = 0$ d'après l'hypothèse de récurrence. On en déduit $(A\mathbf{p}^{k+1}, \mathbf{p}^j) = 0$ si $j < k$. Enfin, dans le cas où $j = k$, avec les expressions de α_k et β_k on déduit

Ceci clôt notre raisonnement par récurrence, et démontre que (7) est vrai pour tout $k \geq 0$.

La première propriété de (7) montre que $\mathbf{r}^j$, $j = 0 \dots k-1$ est une famille de k vecteurs linéairement indépendants appartenant à K_k, c'est donc une base de K_k. On voit donc que la propriété $(\mathbf{r}^k, \mathbf{r}^j) = 0 \quad \forall\, j = 0 \dots k-1$ revient à écrire $(\mathbf{r}^k, \mathbf{v}) = 0$ pour tout $\mathbf{v} \in K_k$. C'est précisément ce qui restait à démontrer pour conclure définitivement la preuve. □

La dimension $\dim(K_k(A; \mathbf{r}_0))$ des espaces de Krylov augmente avec k et on pourrait donc s'attendre à ce que le coût de chaque itération du gradient conjugué augmente avec k. Le résultat précédent démontre que chaque itération du gradient conjugué coûte $\mathcal{O}(n)$. C'est l'une des raisons du succès de la méthode du gradient conjugué.

Algorithme 9 : Gradient Conjugué

$\mathbf{x} = \mathbf{x}^0$
$\mathbf{r} = A\mathbf{x} - \mathbf{b}$
$\mathbf{p} = \mathbf{r}$
tant que $\|\mathbf{r}\| \geq \varepsilon \|\mathbf{r}^0\|$ **faire**
  $\alpha = \frac{(\mathbf{r},\mathbf{r})}{(A\mathbf{p},\mathbf{p})}$
  $\mathbf{x} = \mathbf{x} - \alpha\mathbf{p}$
  $\mathbf{r} = \mathbf{r} - \alpha A\mathbf{p}$
  $\beta = \frac{(\mathbf{r},\mathbf{r})}{\alpha(A\mathbf{p},\mathbf{p})}$
  $\mathbf{p} = \mathbf{r} + \beta\mathbf{p}$
fin

Le gradient conjugué ressemble à la descente de gradient, mais il est bien supérieur en efficacité ! On voit à quel point il est facile d'implémenter le gradient conjugué. La proposition suivante établit un résultat de convergence pour cette méthode itérative.

Theorem 1.75 (Admis). *Soit $A \in \mathbb{M}_{n\times n}$ une matrice symétrique définie positive. On note $\|\mathbf{x}\|_A^2 := (A\mathbf{x}, \mathbf{x})$. Étant donné $\mathbf{x}^0$, $\mathbf{b} \in \mathbb{R}^n$, si $\left(\mathbf{x}^k\right)_{k\geq 0}$ est la suite construite par la méthode du gradient conjugué*

$$\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_A \leq 2\left(\frac{\sqrt{\text{cond}_2(A)}-1}{\sqrt{\text{cond}_2(A)}+1}\right)^k \left\|\mathbf{x}^0 - \mathbf{x}^*\right\|_{\text{A}} \quad \forall k \geq 0$$

1.4.5 Generalized Minimal Residual (GMRes)

La méthode du gradient conjugué (vu dans section précédente) est une méthode de projection orthogonale sur les espaces de Krylov où l'itérée $\mathbf{x}^k \in W_k$ est tel que $\mathbf{r}^k$ est orthogonal à $K_k(A; \mathbf{r}^0)$ ($A = A^t > 0$) :

$$\mathbf{v}^T\left(\mathbf{b} - A\mathbf{x}^{(k)}\right) = 0 \quad \forall\, \mathbf{v} \in K_k\left(A; \mathbf{r}^{(0)}\right)$$

Cette approche est efficace pour la résolution de systèmes symétriques définis positifs. Nous allons maintenant introduire une approche pour le cas général où $\mathbf{x}^k \in W_k$ minimise la norme de résidu $\|\mathbf{r}^k\|_2$:

$$\boxed{} \tag{9}$$

C'est le principe de la méthode Generalized Minimal Residual ou GMRes (cette construction correspond à une projection oblique sur les espaces de Krylov). Afin de préciser la construction de la méthode GMRes, on va chercher à construire une base orthonormale de $K_k(A, \mathbf{r_0}) := K_k$. On sait que $\{A^j\mathbf{r_0}\}_{j=0}^{k-1}$ est une base de K_k mais elle n'est pas orthonormale. On considère la base $\{v_j\}_{j=1}^k$ obtenue à partir de $\{A^j\mathbf{r_0}\}_{j=0}^{k-1}$ suivant un procédé d'orthonormalisation de Gram-Schmidt selon les formules

$$\begin{cases} \mathbf{v}_1 = \mathbf{r_0}/\|\mathbf{r_0}\|_2 \\ \mathbf{w}_{j+1} = A\mathbf{v}_j - \sum_{i=1}^j (A\mathbf{v}_j, \mathbf{v}_i)\mathbf{v}_i = A\mathbf{v}_j - \sum_{i=1}^j \mathbf{v}_i\mathbf{v}_i^t A\mathbf{v}_j \\ \mathbf{v}_{j+1} = \mathbf{w}_{j+1}/\|\mathbf{w}_{j+1}\|_2 \end{cases} \tag{10}$$

On rappelle que la procédure d'orthonormalisation de Gram-Schmidt s'avère instable si l'on ne fait pas attention à la manière dont est calculée concrètement la somme à la deuxième ligne. Le procédé d'Arnoldi, présenté ci-dessous, donne algébriquement le même résultat que la procédure de Gram-Schmidt mais est plus stable numériquement.

Algorithme 10 : Algorithme d'Arnoldi

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}^0$
$\mathbf{v}_1 = \mathbf{r_0}/\|\mathbf{r_0}\|_2$
pour $j = 1$ *à* k **faire**
  $\mathbf{w} = A\mathbf{v}_j$ // Dans G.S. classique on aurait pris $w = A^j\mathbf{r}_0$.
  pour $i = 1$ *à* j **faire**
    $\mathbf{w} = \mathbf{w} - (\mathbf{w}, \mathbf{v}_i)\mathbf{v}_i$
  fin
  $\mathbf{v}_{j+1} = \mathbf{w}/\|\mathbf{w}\|_2$
fin

Introduisons quelques notations. On notera par

$$V_k = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_k]$$

la matrice orthogonale d'ordre $n \times k$ et $\bar{H}_k$ une matrice de Heissenberg de taille $(k+1) \times k$ (une ligne de plus que de colonnes) définie par

C'est à dire, de la forme

$$\bar{H}_k = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & \ldots & h_{1k} \\ h_{21} & h_{22} & h_{23} & h_{24} & \ldots & h_{2k} \\ 0 & h_{32} & h_{33} & h_{34} & \ldots & h_{3k} \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & h_{k,k-1} & h_{k,k} \\ 0 & 0 & 0 & \ldots & 0 & h_{k+1,k} \end{pmatrix}$$

La sous matrice formée des k premières lignes sera notée par H_k. L'intérêt de ces matrices provient du fait qu'elles permettent une écriture matricielle du procédé d'Arnoldi :

$$AV_k = V_{k+1}\bar{H}_k.$$

Par définition de la méthode GMRES on cherche $\mathbf{x}^k \in W_k = \mathbf{x}_0 + K_k$, en choisissant la base $\{\mathbf{v}_j\}_{j=1}^k$ qu'on vient de construire, on réécrit

$$\mathbf{x}^k = \mathbf{x}^0 + V_k\mathbf{y}, \text{ où } \mathbf{y} \in \mathbb{R}^k.$$

On peut donc réécrire le problème de minimisation (9) de manière équivalente :

En résumé, l'algorithme de GMRes à chaque itération k est de la forme :

Algorithme 11 : GMRes Étape k : idée générale

$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}^0$

$\beta = \|\mathbf{r_0}\|_2$

$(V_{k+1}, \overline{H}_k) \leftarrow$ algorithme d'Arnoldi $(A, \mathbf{r_0})$

$\mathbf{y} \leftarrow$ résolution du problème de moindres carrés $(\bar{H}_k, \beta)$ $*$

$\mathbf{x}^k = \mathbf{x}^0 + V_k\mathbf{y}$

Il reste à discuter de la manière dont on peut résoudre le problème de minimisation. Comme $\ker(\bar{H}_k) = \{0\}$, la matrice $\bar{H}_k^t\bar{H}_k$ est symétrique définie positive, et on a une équivalence :

$$\min_{y \in K_k} \|\bar{H}_k\mathbf{y} - \beta\mathbf{e}_1\|_2 \iff \bar{H}_k^t\bar{H}_k\mathbf{y} = \beta\bar{H}_k^t\mathbf{e}_1$$

et on peut appliquer la méthode Cholesky ou la décomposition QR. Notons que compte tenu de la structure de $\bar{H}_k^t$, on a juste à "éliminer" à l'aide d'une matrice orthogonale les termes sous diagonaux pour obtenir R. On procède par la méthode de Givens en utilisant les matrices de rotation plane (voir TD).

Pour garantir une précision satisfaisante, on peut être amené à augmenter k substantiellement, et ceci est réellement problématique car le coût du calcul croît très vite avec k, typiquement en $\mathcal{O}(k^3)$. Cette stratégie telle quelle n'est donc pas raisonnable en général. La solution est d'adapter une variante de la méthode appelé "restarted GMRes".

Résumé : Une méthode de Krylov consiste à construire itérativement (en k) une solution $\mathbf{x}^k \in \mathbf{x}^0 + K_k(A, \mathbf{r}_0)$. Elle est définie par :

- le choix de la base de K_k qu'on utilise.
- le critère d'optimalité qu'on souhaite optimisé.
- La solution de notre problème $\mathbf{x} \in \mathbf{x}^0 + K_{k_0}(A, \mathbf{r}_0)$. Par conséquent, une méthode de Krylov converge en k_0 (critique) itérations. Ces méthodes corres-

pondent donc à des méthodes directes, mais elles sont utilisés comme des méthodes itératives, le but étant de faire moins d'itérations que k_0.

1.4.6 Principe du préconditionnement

L'idée est de résoudre un système équivalent

$$MA\mathbf{x} = M\mathbf{b}.$$

où on souhaite choisir M tel que

$$\mathrm{cond}_2(M^{-1}A) < \mathrm{cond}_2(A).$$

À priori, le meilleur choix de préconditionneur est $M = A^{-1}$ ce qui évidemment n'est pas applicable en pratique car trop coûteux. On construit en général une approximation M de A^{-1} de sorte que le calcul de $MA\mathbf{x}$ soit peu coûteux.

Exemple (Quelques préconditionneurs)**.**
Jacobi : On choisit dans ce cas $M = D^{-1}$ où D est la diagonale de A.
Gauss-Seidel : On choisit dans ce cas $M = (D - E)^{-1}$ (E est la partie triangulaire strictement inférieure de A). ▼

On pourrait penser qu'un tel procédé pose problème pour appliquer à la méthode du gradient conjugué car même si M et A sont symétrique définie positives, il n'y pas de raison que $M^{-1}A$ l'est. Afin de préserver cette propriété on utilise un préconditionnement symétrique :

$$M^tAM\widetilde{x} = M^tb \quad \text{où} \quad \widetilde{x} = M^{-1}x.$$

1.5 Méthode directe ou itérative ?

Tout dépend du problème. En général, les méthodes directes (surtout quand elles sont implémentées de manière sophistiquée) sont plus efficaces que les méthodes itératives quand ces dernières ne sont pas utilisées avec des préconditionneurs performants. Cependant, elles sont plus sensibles au conditionnement de la matrice et peuvent nécessiter une mémoire importante.

Il est également utile de souligner que les méthodes directes ont explicitement besoin des coefficients de la matrice, contrairement aux méthodes itératives. Pour ces dernières, il est seulement nécessaire de pouvoir calculer le produit matrice-vecteur pour des vecteurs arbitraires. Cette propriété est particulièrement intéressante dans les problèmes où la matrice n'est pas construite explicitement.