Chapitre II . Problèmes aux Moindres Carrés Applications

I) Généralités

On considère des systèmes linéaires du type

$$\underset{n}{m\begin{bmatrix} A \end{bmatrix}} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \quad \text{avec} \quad \begin{array}{l} A \in \mathbb{R}^{m \times n} \\ b \in \mathbb{R}^m \\ x \in \mathbb{R}^n \end{array} \quad m \geq n$$

Sauf cas ultra particulier où $b \in \mathrm{Im}(A)$ ce système linéaire sur-déterminé $(m \geqslant n)$ n'admet pas de solution $x \in \mathbb{R}^n$, et on est amené à chercher $\hat{x} \in \mathbb{R}^n$ tel que $A\hat{x}$ s'approche au mieux de $b \in \mathbb{R}^m$.

La notion de "au mieux" peut être traitée en particulier au sens de la norme euclidienne, c'est à dire :

$$\| A\hat{x} - b \|_2 = \min_{x \in \mathbb{R}^n} \| Ax - b \|_2$$

et on parle de solution au sens des moindres carrés.

Illustration sur un exemple :

Soit un mobile libre en chute verticale sous l'action de la pesanteur, dont on mesure les positions à divers instants (hauteur $x(t)$).
On souhaite estimer, à partir des seules observations $\left(t_i, \tilde{x}_i = x(t_i)\right)$, $i = 1, 2 \dots, n$ les valeurs de la position initiale $x_0 = x(0)$

la vitesse initiale $v_0 = \dot{x}(0)$

l'accélération $-g = \ddot{x}(t)$

(le mobile étant supposé chuter sans frottement)

L'équation de la trajectoire en fonction du temps est :

$$x(t) = -\frac{1}{2} g t^2 + v_0 t + x_0$$

On peut donc écrire cette égalité à chaque instant t_i, $i=1\dots m$, qui se traduit matriciellement en fonction des inconnues (paramètres du modèle) sous la forme

$$Av = b \quad , \quad v = \begin{pmatrix} g \\ v_0 \\ x_0 \end{pmatrix} \in \mathbb{R}^3$$

et avec $A = \begin{bmatrix} -\frac{1}{2}t_1^2 & t_1 & 1 \\ \vdots & \vdots & \vdots \\ -\frac{1}{2}t_m^2 & t_m & 1 \end{bmatrix}$ et $b = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_m \end{bmatrix}$

Ainsi, pour estimer (g, v_0, x_0) dans cet exemple on va résoudre le problème d'optimisation

$$\min_{v \in \mathbb{R}^3} \|Av - b\|_2^2$$

Proposition : Soit $A \in \mathbb{R}^{m\times n}$, $m \geq n$, et $b \in \mathbb{R}^m$

- Le problème de moindres carrés :

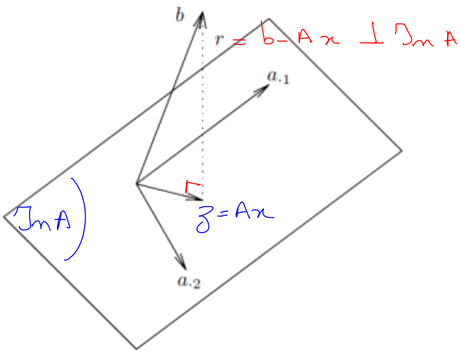
$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$

admet toujours une solution $\hat{x} \in \mathbb{R}^n$.

- Toute solution est caractérisée par le fait qu'elle vérifie le système des équations normales : $A^T A \hat{x} = A^T b$
- La solution $\hat{x} \in \mathbb{R}^n$ est unique ssi $\operatorname{rang}(A) = n$

Interprétation géométrique :

On cherche $z = Ax \in \operatorname{Im} A$ qui est le plus proche du vecteur $b \in \mathbb{R}^m$



$$\boxed{z = \operatorname{Proj}^{\perp}_{\operatorname{Im} A} \cdot b}$$

$z = A\hat{x}$ est caractérisé par $r = b - A\hat{x} \perp \mathrm{Im}\, A$
ou encore $r \in \ker(A^T)$ (norme 2 uniquement) :

$$A^T r = A^T(b - A\hat{x}) = 0$$
$$\Leftrightarrow \boxed{A^T A \hat{x} = A^T b}$$

Si rang $A = n$ est maximal, $A^T A$ est définie positive et la solution $\hat{x}$ du système des équations normales ci-dessus est unique □

Exercice : Soit G symétrique définie positive.
On peut alors considérer le produit scalaire sur $\mathbb{R}^m$, défini par $\langle x, y\rangle_G = x^T G y$
et la norme euclidienne associée

$$\|x\|_G = \sqrt{\langle x, x\rangle_G} = (x^T G x)^{1/2}$$

Q
- Donner la solution du problème aux moindres carrés en norme-G :

$$\hat{x} = \operatorname{Arg}\min_{x \in \mathbb{R}^n} \|Ax - b\|_G$$

où $A \in \mathbb{R}^{m \times n}$, $m \geq n$ et $b \in \mathbb{R}^m$

- Interpréter géométriquement le problème.

II) Méthodes de résolution du problème aux moindres carrés.

Plusieurs techniques sont envisageables :

① Equations normales :

la solution $\hat{x}$ est donnée par $(A^TA)^{-1}A^Tb$
avec A^TA symétrique définie positive (si rang $A = n$)

Une première approche consisterait donc à construire
$S = A^TA$ (la matrice des équations normales)
et à la factoriser par la méthode de Cholesky :

$$A^TA = R^TR$$ avec R triangulaire supérieure

Ensuite, on calcule $\hat{x}$ par les étapes suivantes

ⓐ $z = A^Tb$ (produit matrice-vecteur)

ⓑ $y = R^{-T}z$ (forward-substitution)

ⓒ $\hat{x} = R^{-1}y$ (backward substitution)

Le coût en nombre d'opérations est : (A pleine)
$$mn^2 + n^3/3$$

② Factorisation QR

On peut factoriser A sous forme QR

- soit avec une factorisation à base de transformées de Householder, qui nous donne :

$$m\underset{n}{\boxed{A}} = m\underset{m}{\boxed{Q}}\begin{bmatrix} R \\ \emptyset \end{bmatrix}$$

où Q est un produit de n réflexions de Householder

- soit à l'aide de l'algorithme de Gran-Schmidt modifié, qui donne une factorisation réduite :

$$m \underset{n}{\boxed{A}} = m \underset{n}{\boxed{Q_1}} \boxed{R}$$

La résolution se fait alors avec les étapes suivantes :

ⓐ $z = Q^T b$
et z_1 correspond aux n 1ères composantes de z

ⓐ $z_1 = Q_1^T b$

(exo)

ⓑ $\hat{x} = R^{-1} z_1$ (backward substitution)

Il est à noter que le stockage de la matrice Q en mémoire n'est pas nécessaire - Il est juste utile de garder en mémoire les vecteurs v_k définissant les réflexions de Householder, et ceci peut se faire dans un volume mémoire restreint.
En outre, si on n'a qu'un seul système à résoudre on peut directement appliquer les réflexions de Householder au vecteur b "au fil de l'eau" et construire $z = Q^T b$ directement en même temps qu'on construit R.

Le coût en nombre d'opérations est : (QR Householder)
$2mn^2 - 2n^3/3$

Dans le cas où on aurait une séquence de problèmes de moindres carrés à résoudre, avec la même matrice A et des seconds-membres b qui changent, il est possible de considérer l'approche appelée :
Equations semi-normales

l'idée est de faire une facto QR de A (Householder QR) et de ne garder que R en mémoire, puis d'appliquer les étapes (a) (b) (c) spécifiques aux équations normales pour calculer $\hat{x}$.

En effet $\quad A^TA = [R^T \emptyset] \underbrace{Q^TQ}_{I_m} \begin{bmatrix} R \\ \emptyset \end{bmatrix} = R^TR$

et donc les facteurs de Choleski de A^TA correspondent à la matrice R de la facto QR.

La différence fondamentale entre la méthode des équations normales et celle des équations semi-normales réside dans la construction de R, qui change en arithmétique finie.

③ Système augmenté

Il est possible aussi de résoudre le système dit "système augmenté" suivant :

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

En effet, cela revient à poser $r = b - Ax$ et à écrire la condition d'orthogonalité : $r \in \operatorname{Ker} A^T$

Ce système est aussi équivalent aux conditions nécessaires et suffisantes du problème d'optimisation strictement convexe suivant :

$$\begin{cases} \operatorname{Min} \frac{1}{2} \| r - b \|_2^2 \\ r \,/\, A^T r = 0 \end{cases} \qquad \text{(exo)}$$

dans lequel x joue en fait le rôle des paramètres de Lagrange.

Le système augmenté ci-dessus peut être factorisé par une technique de type LDL^T avec pivotage. Ceci peut être avantageux dans les cas particuliers où la matrice A possède des lignes relativement denses, car alors les facteurs R de la facto QR seront aussi pleins (ou presque pleins). Une permutation symétrique du système augmenté permet en effet de "rejeter" à la fin ces lignes presque pleines dans l'ordre des pivots de Gauss, et fournir des facteurs L relativement creux en comparaison des facteurs R dans la factorisation QR.

Rque : si, dans la factorisation de Gauss du système augmenté, on commence par pivoter sur les m 1ers éléments diagonaux (ceux de I_m dans le bloc $(1,1)$) alors on construit explicitement $A^T A$ dans le complément de Schur du bloc $(2,2)$
[à éviter]

4 SVD et pseudo-inverse

Au chapitre I, on a introduit la notion de pseudo-inverse de Moore-Penrose, qui peut être décrite à l'aide de la décomposition en valeurs singulières de A :

$$\underset{m\times n}{A} = \underset{m\times m}{U}\begin{bmatrix}\Sigma \\ \emptyset\end{bmatrix}\underset{n\times n}{V^T}$$

où U et V sont unitaires, et Σ avec les valeurs singulières σ_i dans l'ordre décroissant sur la diagonale.

La pseudo inverse de Moore Penrose de A s'écrit alors

$$A^+ = V \begin{bmatrix}\Sigma^{-1} & \emptyset\end{bmatrix} U^T$$

sous l'hypothèse que $\operatorname{rang} A = n$ et donc que les σ_i sont toutes non nulles. $(\sigma_i > 0,\ 1 \leq i \leq n)$

On a aussi vu (en exo) que $P = AA^+$ est le projecteur orthogonal sur $\operatorname{Im} A$, et on a donc

$$z = \operatorname{Proj}^{\perp}_{\operatorname{Im} A} b = AA^+ b = A\hat{x}$$

d'où

$$\hat{x} = A^+ b = V \Sigma^{-1} U_1^T b$$

En pratique, la SVD étant coûteuse à calculer, cette méthode de résolution du problème aux moindres carrés est peu utilisée.

Par contre, dans le cas particulier où A n'est pas de rang maximal ($\operatorname{rang} A = r < n$), la solution $\hat{x}$ du problème aux moindres carrés n'est plus unique, et on peut être amené alors à considérer la solution dite "solution de norme minimale" du problème aux moindres carrés, définie par :

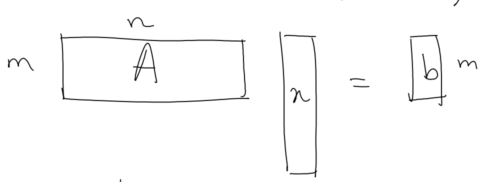
$$\min_{x \in \mathcal{L}} \|x\|_2 \quad \text{avec} \quad \mathcal{L} = \{ x \;/\; \|Ax-b\|_2 \text{ est minimale} \}$$

La solution $\hat{x}$ de ce problème peut être calculée à l'aide de la SVD de A :

$$\hat{x} = V \begin{pmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T b$$

$$= V_r \Sigma_r^{-1} U_r^T b \qquad \text{(exo)}$$

De même, dans le cas des problèmes dits "sous déterminés", dans lesquels il y a plus de paramètres que d'observations ($m > n$)



On peut aussi être amené à considérer la solution de norme minimale définie par le problème d'optimisation :

$$\begin{cases} \min \|x\|_2 \\ x \;/\; Ax = b \end{cases}$$

qui, dans le cas où A est de rang maximal égal à m est donné par

$$\hat{x} = A^T (AA^T)^{-1} b \qquad \text{(exo)}$$

$$= V \begin{bmatrix} \Sigma^{-1} \\ 0 \end{bmatrix} U^T b = V_1 \Sigma^{-1} U^T b$$

III) Comparaison entre les équations normales et la méthode par factorisation QR

En arithmétique finie (présence d'erreurs d'arrondi), les deux méthodes de résolution des problèmes aux moindres carrés, à savoir "equations normales" ou "facto. QR" se comportent de manière très différente.

ⓐ Tout d'abord, la construction de la matrice $C = A^TA$ dans les équations normales peut présenter des problèmes dits de "cancellation", et conduire à une matrice presque singulière, voire non définie positive.

Par exemple, pour $A = \begin{pmatrix} 1 & 1 \\ 0 & \sqrt{\varepsilon}/10 \\ \sqrt{\varepsilon}/10 & 0 \end{pmatrix}$ (où ε = eps. mach)

on aurait : $C = A^TA = \begin{bmatrix} 1+\frac{\varepsilon}{100} & 1 \\ 1 & 1+\frac{\varepsilon}{100} \end{bmatrix} =$

mais comme $1+\frac{\varepsilon}{100} = 1$ en arithmétique finie,

$C = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ sur machine, qui est singulière.

$\Rightarrow$ la factorisation de cholesky de C échouera.

ⓑ Les erreurs d'analyse de perturbation à posteriori vont porter sur des données de type différent entre les deux méthodes, ce qui peut rendre difficile la comparaison entre les résultats obtenus par chacune des deux méthodes.

Dans ce qui suit, on désignera par c_{mn} (ou c_n) une constante générique qui augmente faiblement avec m et n, et on définit par $\text{cond}_2(A) = \|A\|_2\|A^+\|_2 = \frac{\sigma_1}{\sigma_n}$ le conditionnement de A en norme 2.

Pour ce qui est de l'analyse d'erreur à posteriori, on peut citer les deux résultats suivants.

Stabilité inverse de la méthode QR : (Wilkinson 1965)

Supposons que la solution approchée $\tilde{x}$ de $\min_x \|Ax-b\|_2$ ait été obtenue par factorisation QR de A (Givens / Householder), A étant supposée de rang maximal égal à n.

Supposons de plus que A est telle que $c_{mn}\,\varepsilon\,\mathrm{cond}_2(A) < 1$.
Alors, sous ces hypothèses, il existe une matrice E et un vecteur f tels que $\tilde{x}$ soit une solution exacte du problème

$$\min_x \|(A+E)x-(b+f)\|_2 \quad \text{avec} \begin{cases} \|E\|_F \leq c_{mn}\,\varepsilon\,\|A\|_F \\ \|f\|_2 \leq c_{mn}\,\varepsilon\,\|b\|_2 \end{cases}$$

Stabilité inverse de la méthode des équations normales :

Supposons que la solution approchée $\tilde{x}$ de $\min_x \|Ax-b\|_2$ ait été obtenue par factorisation de Choleski des équations normales, avec $C = A^T A$ définie positive.
Supposons de plus que A est telle que $c_{mn}\,\varepsilon\,\mathrm{Cond}_2(A)^2 < 1$.
Sous ces hypothèses (la factorisation de Choleski est garantie de s'achever sans problème), il existe une matrice ΔC telle que $\tilde{x}$ soit une solution exacte du problème

$$(C+\Delta C)\tilde{x} = d \quad (= A^T b)$$

$$\text{avec } \|\Delta C\|_2 \leq c_{mn}\,\varepsilon\,\|A\|_2^2 \left(1 + \frac{\|b\|_2}{\|A\|_2\,\|\tilde{x}\|_2}\right) + O(\varepsilon^2)$$

Pour ce qui est de l'analyse de l'erreur sur la solution calculée, on peut établir le résultat suivant :

Majoration de l'erreur directe :

Soit x^* la solution exacte du problème de moindres carrés $\min_x \|Ax-b\|_2$, avec A de rang maximal égal à n.

① Supposons que $\tilde{x} = x^* + \Delta x$ soit la solution calculée obtenue par factorisation QR (Householder/Givens) de A, avec $c_{mn}\,\varepsilon\,\mathrm{cond}_2(A) < 1$, alors l'erreur sur la solution vérifie :

$$\frac{\|\Delta x\|_2}{\|x^*\|_2} \leq c_{mn}\,\mathrm{Cond}_2(A) \left(1 + \frac{\|b\|_2}{\|A\|_2\,\|x^*\|_2} + \mathrm{cond}_2(A)\frac{\|r^*\|_2}{\|A\|_2\,\|x^*\|_2}\right)\varepsilon$$

$$(\text{où } r^* = b - Ax^*)$$

② Supposons que $\tilde{x} = x^* + \Delta x$ soit la solution calculée obtenue par factorisation de Choleski des équations normales, avec A telle que $c_{mn}\,\varepsilon\,\mathrm{cond}_2(A)^2 < 1$, alors l'erreur sur la solution vérifie

$$\frac{\|\Delta x\|_2}{\|x^*\|_2} \leq c_{mn}\,\mathrm{Cond}_2(A)^2 \left(1 + \frac{\|b\|_2}{\|A\|_2\,\|\tilde{x}\|_2}\right)\varepsilon$$

Interprétation du résultat:

- Pour un problème où $\frac{\|r\|_2}{\|A\|_2 \|x\|_2}$ est faible, la méthode QR ne fait pas intervenir $\mathrm{Cond}_2(A)^2$, et sera donc plus précise que la méthode basée sur la factorisation de Cholesky de $C = A^T A$.
- Dans le cas contraire, tous les cas de figure sont envisageables.
- Dans la littérature, la méthode QR a la réputation d'être la plus précise des deux.
- La méthode QR (Householder/Givens) s'applique potentiellement à une classe de matrices plus large ($C_{mn} \varepsilon \, \mathrm{Cond}_2(A) < 1$ → sans le carré du conditionnement)
- Remarquons aussi que, pour $m \gg n$, la méthode QR est potentiellement deux fois plus coûteuse que l'approche par les équations normales.

Pour finir, si on est particulièrement intéressés par la base orthonormée de $\mathrm{Im}(A)$ donnée par les colonnes de Q (GS, MGS, QR-Householder, QR-Givens), les pertes d'orthogonalités seront bien moindres (en arithmétique finie) avec les factorisations QR à base de transformées de Householder ou de Givens.

IV) Méthodes de Krylov pour la résolution des systèmes linéaires

Dans le cas des problèmes de grande taille, on peut être amené (pour des raisons de coût en calcul et/ou en mémoire) à considérer des méthodes de résolution itératives, qui permettent de construire une suite d'itérés $x^{(k)}$ convergeant graduellement vers la solution x^*, sans avoir à factoriser explicitement la matrice A considérée.

En outre, avec des critères d'arrêt appropriés, on peut aussi stopper les itérations à un niveau d'approximation qui sera "acceptable" relativement au problème considéré et aux données en entrée

L'objectif de cette partie est de présenter une des grandes classes de méthodes itératives, à savoir donc celle des "méthodes de Krylov"

Définition

Pour une matrice carrée inversible, on définit l'espace de Krylov d'ordre k associé à A et b, par :

$$\mathcal{K}(A, b, k) = \text{Vect}\left\{b, Ab, A^2b, \dots, A^{k-1}b\right\}$$

Rque : considérons le polynôme caractéristique de A

$$\det(A - XI_n) = P_n(X) = \sum_{k=0}^{n} \alpha_k X^k$$

avec $\alpha_0 = \det A = P_n(0)$
et $\alpha_0 \neq 0$ ssi A inversible

et comme $P_n(A) = 0 = \alpha_0 I_n + \alpha_1 A + \dots + \alpha_n A^n$
$P_n(A)b = 0 = \alpha_0 b + \alpha_1 Ab + \dots + \alpha_n A^n b$

$$\Rightarrow \quad b = -\frac{1}{\alpha_0}\left(\alpha_1 Ab + \dots + \alpha_n A^n b\right) \quad (A \text{ inversible})$$

$$= A\Big[\underbrace{-\frac{1}{\alpha_0}\left(\alpha_1 b + \dots + \alpha_n A^{n-1} b\right)}_{x \,\in\, \mathcal{K}(A,b,n)}\Big]$$

(ceci est aussi vrai avec le polynôme minimal, de $d^o \leq n$)

L'idée sous-jacente des méthodes de Krylov est donc de construire itérativement une base de $\mathcal{K}(A,b,k)$, $k=1,2,\dots$ et de décomposer la solution $x^{(k)}$ dans cette base.

Les méthodes diffèrent essentiellement dans la façon de construire cette base, ainsi que dans la manière de déterminer l'itéré $x^{(k)}$ comme combinaison linéaire des vecteurs de la base ainsi déterminée à l'itération k.

On peut citer, par exemple :

- L'approche Ritz-Galerkin : $x^{(k)}$ est choisi tq $b - Ax^{(k)} \perp \mathcal{K}(A,b,k)$
- le résidu minimum : trouver $x^{(k)} \in \mathcal{K}(A,b,k)$ tq $\|b - Ax^{(k)}\|_2$ ~~soit~~ minimal
- l'approche Petrov-Galerkin
- l'erreur minimale

① Construction de la base de $\mathcal{K}(A,b,k)$

Une première idée consiste simplement à déterminer itérativement une base orthonormée de l'espace $\mathcal{K}(A,b,k)$. Pour cela, on peut naturellement utiliser le procédé d'orthogonalisation de Gram-Schmidt :

Méthode d'Arnoldi

On pose $\beta_0 = \|b\|_2$ et $v_0 = \dfrac{b}{\beta_0}$

Pour $k = 1, 2, \dots, m-1$, faire

  Calculer $h_{ik} = v_i^T A v_k$ $1 \leq i \leq k$

  Calculer $w_k = Av_k - \sum_{i=1}^{k} h_{ik} v_i$ (CGS)

  Poser $h_{k+1,k} = \|w_k\|_2$

  Si $h_{k+1,k} = 0$, arrêt du processus

  Sinon, poser $v_{k+1} = \dfrac{w_k}{h_{k+1,k}}$

Fin Pour

Propriétés :

- Les vecteurs v_j, $j = 1 \dots k$ forment une base orthonormée (en arithmétique exacte) de $\mathcal{K}(A, b, k)$

- A chaque itération k, on a :

$$A V_k = V_{k+1} H_{k+1,k} \quad \text{avec}$$

$$V_k = \left[v_1 \text{---} v_k \right] \quad \text{et} \quad V_{k+1} = \left[V_k, v_{k+1} \right]$$

$H_{k+1,k} \in \mathbb{R}^{(k+1) \times k}$ avec $H_{k+1,k}$ Hessenberg supérieure

$$H_{k+1,k} = \begin{bmatrix} h_{11} & h_{12} & \text{---} & h_{1k} \\ h_{21} & h_{22} & \text{---} & h_{2k} \\ & h_{32} & \ddots & \vdots \\ & \emptyset & \ddots & h_{kk} \\ & & & h_{k+1,k} \end{bmatrix}$$

- Si $h_{k+1,k} = 0$, alors $A V_k = V_k H_{k,k}$

avec $H_{k,k} \in \mathbb{R}^{k \times k}$ carrée Hessenberg supérieure
et toute valeur propre de $H_{k,k}$ est aussi valeur propre de A

- Si $h_{k+1,k} = 0$ à l'itération k, alors la solution x du système linéaire $Ax = b$ appartient à l'espace de Krylov $\mathcal{K}(A, b, k-1)$

- Chaque itération implique $O(kn)$ calculs.

③ Calcul de l'itéré $x^{(k)} \in \mathcal{K}(A, b, k) = \mathrm{Im}(V_k)$

Posons $x^{(k)} = V_k y^{(k)} \in \mathcal{K}(A, b, k)$
on peut chercher à déterminer $y^{(k)}$ de façon à minimiser $\| b - A x^{(k)} \|_2^2$ (ce qui est dans l'esprit des moindres carrés)

On a : $\| b - A x^{(k)} \|_2^2 = \| b - A V_k y^{(k)} \|_2^2$

$$= \| b - V_{k+1} H_{k+1,k} y^{(k)} \|_2^2$$

(Pythagore) $$= \| \underbrace{V_{k+1} V_{k+1}^T b - V_{k+1} H_{k+1,k} y^{(k)}}_{\in \mathrm{Im}(V_{k+1})} \|_2^2$$

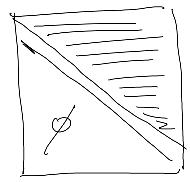
$$+ \underbrace{\| b - V_{k+1} V_{k+1}^T b \|_2^2}_{\perp \text{ à } \mathrm{Im}(V_{k+1})}$$

et par conséquent, minimiser $\| b - A x^{(k)} \|_2^2$ en fonction de $x^{(k)} \in K(A, b, k)$ revient à minimiser en fonction de $y^{(k)} \in \mathbb{R}^k$:

$$\| V_{k+1} (V_{k+1}^T b - H_{k+1,k} \, y^{(k)}) \|_2^2 = \underline{\| \beta_0 e_1 - H_{k+1,k} \, y^{(k)} \|_2^2}$$

puisque les colonnes de V_{k+1} sont orthonormées et que $b = v_1 \times \beta_0$. ($1^{ère}$ colonne de V_{k+1})

On a donc au final à résoudre un problème aux moindres carrés réduit, avec un second-membre très particulier (fixé dès la $1^{ère}$ itération par β_0) et une matrice $H_{k+1,k}$ rectangulaire de taille $k+1 \times k$, Hessenberg supérieure

$H_{k+1,k} =$ 

L'idée, c'est alors d'utiliser des rotations de Givens pour annuler en $O(k)$ opérations la sous-diagonale de $H_{k+1,k}$, et d'obtenir à moindre coût une factorisation QR de $H_{k+1,k}$:

$$H_{k+1,k} = Q^{(k+1)} \begin{bmatrix} R_{k \times k} \\ 0 \end{bmatrix}$$

où $Q^{(k+1)} = \prod_{j=k}^{1} G_j$ (G_j rotation de Givens) peut être construite de manière récursive (1 rotation seulement par itération)

Propriété : Il est possible de vérifier que :

$$\| b - A x^{(k)} \| = h_{k+1,k} \quad (\text{exo})$$

Dans le cas de la résolution de systèmes linéaires, ceci peut donner un estimateur à moindre coût de la convergence.

Remarque : Pour des raisons de perte d'orthogonalité on préfère en général utiliser le processus d'orthogonalisation de Gram-Schmidt modifié, ce qui donne l'algorithme GMRES_MGS.

Algorithme GMRES_MGS (Generalised minimum residual with modified Gram_Schmidt orthogonalisation)

Input : $A \in \mathbb{R}^{n \times n}$ inversible
$b \in \mathbb{R}^n$
$x^{(0)} \in \mathbb{R}^n$ itéré initial ($= \emptyset$, sans a priori)

Initialisation : Poser
$r_0 = b - A x^{(0)}$
$h_{10} = \|r_0\|_2$
$k = 0$

Tant que $h_{k+1,k} > 0$ (et que $k \leq m$)
- $v_{k+1} = r_k / h_{k+1,k}$
- $k = k+1$
- $r_k = A v_k$
- Pour $i = 1 : k$ } (MGS)
  - $h_{ik} = v_i^T r_k$
  - $r_k = r_k - h_{ik} v_i$
- fin Pour
- $h_{k+1,k} = \|r_k\|_2$
- $x^{(k)} = x^{(0)} + V_k y^{(k)}$ où $y^{(k)} = \mathrm{ArgMin} \| h_{10} e_1 - H_{k+1,k} y^{(k)} \|_2^2$

fin Tantque
Retourner $x = x^{(k)}$.

En pratique on arrête l'algo. sur une tolérance donnée et un nb_max d'itérations.

③ Cas où A est symétrique : La méthode de Lanczos

Dans le cas où A est symétrique $A = A^T$
le procédé d'Arnoldi se simplifie, car

$$V_k^T A V_k = H_{kk} = (V_k^T A V_k)^T = H_{kk}^T$$

et donc $H_{k,k}$ devient tridiagonale symétrique
(Hessenberg supérieure et symétrique)

Procédé de Lanczos : (A symétrique inversible)

Poser $v_0 = 0$ (fictif - pour la $1^{ère}$ itération)
$\beta_0 = \|b\|_2$, $v_1 = b/\|b\|$

Pour $k = 1, 2, \dots$ ($\Leftrightarrow$ Arnoldi simplifié dans le cas symétrique)

$$\alpha_k = v_k^T A v_k$$
$$r_k = A v_k - \alpha_k v_k - \beta_{k-1} v_{k-1}$$
$$\beta_k = \|r_k\|_2$$
$$v_{k+1} = r_k / \beta_k$$

Fin Pour

L'algorithme GMRES, dans le cas symétrique,
devient L'algorithme MINRES

4 Cas où A est symétrique définie positive : L'algorithme du Gradient Conjugué

(Construction détaillée en TD)

Algorithme du Gradient Conjugué

Input : A SPD

$b \in \mathbb{R}^n$

$x^{(0)} \in \mathbb{R}^n$ itéré initial

Initialisation : Poser $r^{(0)} = b - A x^{(0)}$ et $p^{(0)} = r^{(0)}$

Pour $k = 1, 2 \dots$ jusqu'à convergence

$$\alpha_{k-1} = \frac{r^{(k-1)^T} r^{(k-1)}}{p^{(k-1)^T} A p^{(k-1)}}$$

$$x^{(k)} = x^{(k-1)} + p^{(k-1)} \alpha_{k-1}$$

$$r^{(k)} = r^{(k-1)} - A p^{(k-1)} \alpha_{k-1}$$

$$\beta_k = \frac{r^{(k)^T} r^{(k)}}{r^{(k-1)^T} r^{(k-1)}}$$

$$p^{(k)} = r^{(k)} + p^{(k-1)} \beta_k$$

Fin Pour

Pour la convergence, on peut monitorer la valeur de $\dfrac{\| r^{(k)} \|_2}{\| r^{(0)} \|_2}$

ou bien encore $\dfrac{\| r^{(k)} \|_2}{\|A\|_2 \, \|x^{(k)}\|_2 + \|b\|_2}$ (erreur inverse)

On peut aussi citer le résultat d'analyse de convergence suivant :

Proposition :

Soit $x^{(k)}$ le k ième itéré dans l'algorithme du Gradient Conjugué, et $x^* = A^{-1} b$ (la solution exacte).
On a alors :

$$\| x^{(k)} - x^* \|_A \leq 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \| x^{(0)} - x^* \|_A$$

où $\kappa_2(A)$ est le conditionnement de A en norme 2.

(Démo. en exo)

Le résultat précédent montre que la convergence peut être améliorée quand le conditionnement de la matrice d'itération est peu élevé, et il est souvent utile de préconditionner le système linéaire :

Le préconditionnement, dans le cas de l'algorithme du Gradient conjugué doit garantir l'équivalence avec un système symétrique défini positif (SPD)

Soit donc M une matrice de préconditionnement SPD et $M = C^T C$ sa factorisation de Choleski.

Le système linéaire préconditionné par M devient :

$$\underline{M^{-1} A x = M^{-1} b}$$

qui est aussi équivalent à

$$C^{-1}(C^{-T} A C^{-1}) C x = C^{-1} C^{-T} b$$

ou encore

$$C^{-T} A C^{-1} \tilde{x} = C^{-T} b = \tilde{b}$$

et c'est sur ce système qu'on va pouvoir appliquer la méthode du gradient conjugué, car la matrice

$$\underline{\tilde{A} = C^{-T} A C^{-1}} \text{ est SPD}$$

$$(\text{avec } \tilde{x} = Cx \text{ et } \tilde{b} = C^{-T} b)$$

$\tilde{A}$ est similaire à $M^{-1}A$ et $\kappa(M^{-1}A)$ peut être grandement amélioré si $M \simeq A$

On peut vérifier (en exo) que le CG appliqué au système modifié ci-dessus, devient en utilisant les changements de base inverses qui lient $\tilde{x}$ à x et $\tilde{b}$ à b :

Algorithme du PCG

Input : A et M, SPD
$b \in \mathbb{R}^n$, et $x^{(0)} \in \mathbb{R}^n$ itéré initial

Initialisation : $r^{(0)} = b - A x^{(0)}$
$z^{(0)} = M^{-1} r^{(0)}$, $p^{(0)} = z^{(0)}$

Pour $k = 1, 2 \dots$ jusqu'à convergence

$$\alpha_{k-1} = \frac{r^{(k-1)T} z^{(k-1)}}{p^{(k-1)T} A p^{(k-1)}}$$

$$x^{(k)} = x^{(k-1)} + p^{(k-1)} \alpha_{k-1}$$

$$r^{(k)} = r^{(k-1)} - A p^{(k-1)} \alpha_{k-1}$$

$$z^{(k)} = M^{-1} r^{(k)}$$ (utiliser les facteurs de Cholesky de M)

$$\beta_k = \frac{r^{(k)T} z^{(k)}}{r^{(k-1)T} z^{(k-1)}}$$

$$p^{(k)} = z^{(k)} + p^{(k-1)} \beta_k$$

Fin Pour

V) Méthodes de Krylov pour la résolution des problèmes de moindres carrés

1) Application du CG pour résoudre un problème de MdC

Soit $\underset{m}{\overset{n}{\boxed{A}}}$ $(m \geq n)$ de rang n.

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 \iff A^T A x = A^T b$$

et A étant de rang maximal égal à n, $A^T A$ est SPD.
On peut donc naturellement penser à utiliser le CG pour résoudre itérativement le système des équations normales.

Ceci donne lieu à un algorithme spécifique : CGNR (Conjugate Gradient to minimize the Norm of the Residual)

Algorithme CGNR :

Input : A, b, $x^{(0)}$ (A rectangulaire $m \times n$, $m \geqslant n$)

Init : $r^{(0)} = b - A x^{(0)}$

$g^{(0)} = A^T r^{(0)}$, $p^{(0)} = g^{(0)}$

Pour $k = 1, 2, \ldots$ jusqu'à convergence

$$\alpha_{k-1} = \frac{g^{(k-1)^T} g^{(k-1)}}{(A g^{(k-1)})^T (A g^{(k-1)})}$$

$$x^{(k)} = x^{(k-1)} + p^{(k-1)} \alpha_{k-1}$$

$$r^{(k)} = r^{(k-1)} - A p^{(k-1)} \alpha_{k-1}$$

$$g^{(k)} = A^T r^{(k)}$$

$$\beta_k = \frac{g^{(k)^T} g^{(k)}}{g^{(k-1)^T} g^{(k-1)}}$$

$$p^{(k)} = g^{(k)} + p^{(k-1)} \beta_k$$

Fin Pour

(Regarder aussi CGNE)

$$Ax = b \iff \begin{cases} A A^T y = b & \leftarrow \text{CG} \\ x = A^T y \end{cases}$$

Propriété : A chaque itération, CGNR calcule un itéré $x^{(k)}$ qui minimise $\| Ax - b \|_2^2$

pour $x \in x^{(0)} + \mathcal{K}(A^TA, A^T r^{(0)}, k)$

(où $r^{(0)} = b - A x^{(0)}$)

(vérification en exo)

② Procédé de bidiagonalisation de Lanczos-Golub-Kahan

Pour résoudre le problème de MdC $\min_x \| Ax - b \|_2^2$ on peut construire deux base orthonormées associées aux deux espaces de Krylov suivants :

espace dit "à gauche" $\mathcal{K}(AA^T, b, k) = \operatorname{Im} U_k$

espace dit "à droite" $\mathcal{K}(A^TA, A^Tb, k) = \operatorname{Im} V_k$

Posons $u^{(1)} = \dfrac{b}{\|b\|} = \dfrac{b}{\beta_1}$ $(\beta_1 = \|b\|)$ et $v^{(1)} = \dfrac{A^T u^{(1)}}{\|A^T u^{(1)}\|_2} = \dfrac{A^T u^{(1)}}{\alpha_1}$

Puis, en considérant de manière récurrente que

$U_k = [u^{(1)} \dots u^{(k)}]$, $V_k = [v^{(1)} \dots v^{(k)}]$ à l'étape k

$$\begin{cases} p^{(k+1)} = A v^{(k)} - U_k U_k^T A v^{(k)} \quad \text{et} \quad u^{(k+1)} = \dfrac{p^{(k+1)}}{\|p^{(k+1)}\|_2} = \dfrac{p^{(k+1)}}{\beta_{k+1}} \\[2ex] q^{(k+1)} = A^T u^{(k+1)} - V_k V_k^T A^T u^{(k+1)} \quad \text{et} \quad v^{(k+1)} = \dfrac{q^{(k+1)}}{\|q^{(k+1)}\|_2} = \dfrac{q^{(k+1)}}{\alpha_{k+1}} \end{cases}$$

Propriétés :

$$p^{(k+1)} = A v^{(k)} - u^{(k)} u^{(k)T} A v^{(k)}$$

$$q^{(k+1)} = A^T u^{(k+1)} - v^{(k)} v^{(k)T} A u^{(k+1)}$$

et $u^{(k)T} A v^{(k)} = \alpha_k$, $v^{(k)T} A u^{(k)} = \beta_{k+1}$

(en exo)

Ceci conduit à :

Procédé de bidiagonalisation de L.G.K

$\beta_1 u^{(1)} = b$, $\alpha_1 v^{(1)} = A^T u^{(1)}$

Pour $k = 1 \dots n$

$$\beta_{k+1} u^{(k+1)} = A v^{(k)} - u^{(k)} \alpha_k \qquad \left(\beta_{k+1} = \| A v^{(k)} - u^{(k)} \alpha_k \|_2\right)$$

$$\alpha_{k+1} v^{(k+1)} = A^T u^{(k+1)} - v^{(k)} \beta_{k+1} \qquad \left(\alpha_{k+1} = \| A^T u^{(k+1)} - v^{(k)} \beta_{k+1} \|\right)$$

Fin Pour

avec les α_j et β_j déterminés de sorte que $\| u^{(j)} \|_2 = \| v^{(j)} \|_2 = 1$

Proposition : Notons $U_k = [u^{(1)} \dots u^{(k)}]$

$V_k = [v^{(1)} \dots v^{(k)}]$

et $B_{k+1,k} = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \alpha_3 & \\ & & \ddots & \ddots \\ & & & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}$ bidiagonale, de taille $(k+1 \times k)$.

On a alors :

$$A V_k = U_{k+1} B_{k+1,k}$$

$$A^T U_{k+1} = V_{k+1} B^T_{k+1,k}$$

(Vérification en exo) $\Rightarrow U^T_{k+1} A V_k = B_{k+1,k}$

Pour résoudre de manière approchée le problème

$$\min_x \| Ax - b \|_2^2$$

on peut chercher $x^{(k)} \in \operatorname{Im} V_k$ et résoudre alors le problème

$$\min_{y \in \mathbb{R}^k} \| A V_k y - b \|_2^2$$

Ceci revient à minimiser $\| U_{k+1} B_{k+1,k}\, y - U_{k+1} U_{k+1}^T b \|_2^2$

$$+ \; \underbrace{\| (I - U_{k+1} U_{k+1}^T)\, b \|_2^2}_{(\text{fixe / à } y)}$$

La norme euclidienne étant invariante unitairement, on est ramené au problème

$$\underset{y \in \mathbb{R}^k}{\mathrm{Min}} \; \| B_{k+1,k}\, y - \beta_1 e_1 \|_2^2$$

qui peut se résoudre là encore très facilement à l'aide de rotations de Givens pour réaliser une facto QR de la matrice bidiagonale $B_{k+1,k}$.

Il est à noter que les rotations de Givens appliquées à $B_{k+1,k}$ pour annuler la sous-diagonale, donneront une matrice factorisée R qui sera triangulaire supérieure bidiagonale elle aussi.

Il en résulte des recurrences à 2 termes seulement pour la mise à jour de la solution de manière itérative.

Sans détailler tous les aspects techniques (c.f. littérature) cela donne lieu à l'algorithme LSQR de Paige et Saunders.