MDI 210 - Optimisation

1 Problèmes de l'analyse numérique

On considère un système linéaire écrite sous la forme matricielle $Ax = b$.
Deux types d'erreurs peuvent être commises :

- Les erreurs d'arrondi, dues aux limites du codage employé.
- Les erreurs de troncature, due à la limite choisie en nombre d'iterations.

1.1 Conditionnement d'un système linéaire

Def. On appelle **conditionnement** de A (relativement à la norme $\|\cdot\|$, la quantité $\|A\| \cdot \left\|A^{-1}\right\|$, que l'on note $\mathrm{cond}_{\|\cdot\|}(A)$ ou $\mathrm{cond}(A)$.

Dans le cas $A(x+\delta x) = b + \delta b$ on a alors $\frac{\|\delta x\|}{\|x\|} \leqslant \mathrm{cond}(A)\frac{\|\delta b\|}{\|b\|}$. Dans le cas $(A+\delta A)(x+\delta x) = b$ alors $\frac{\|\delta x\|}{\|x+\delta x\|} \leqslant \mathrm{cond}(A)\frac{\|\delta A\|}{\|A\|}$.

Une matrice est d'autant mieux conditionnée que son conditionnement est proche de 1.

Th. *Soit A une matrice inversible. On a alors :*

- $\mathrm{cond}(A) \geqslant 1$
- $\mathrm{cond}(A) = \mathrm{cond}(A^{-1})$
- $\forall \alpha \neq 0, \mathrm{cond}(\alpha A) = \mathrm{cond}(A)$
- *En notant cond_2 le conditonnement associé à $\|\cdot\|_2$ et en notant respectivement $\mu_1(A)$ et $\mu_n(A)$ la plus petite et la plus grande des valeurs singulières de A,* $\mathrm{cond}_2(A) = \frac{\mu_n(A)}{\mu_1(A)}$.
- *Si A est normale (i.e. $AA^* = A^*A$),* $\mathrm{cond}_2(A) = \frac{\max_i|\lambda_i(A)|}{\min_i|\lambda_i(A)|}$ *où les $\lambda_i(A)$ représentent les valeurs propres de A.*
- *Si A est unitaire ou orthogonale,* $\mathrm{cond}_2(A) = 1$.
- $\mathrm{cond}_2(A)$ *est invariant par transformation unitaire ou orthogonale : si $UU^* = I$ (resp. $U^tU = I$) alors*

$$\mathrm{cond}_2(A) = \mathrm{cond}_2(AU) = \mathrm{cond}_2(UA) = \mathrm{cond}_2(U^*AU) \qquad (\textit{resp. } \mathrm{cond}_2({}^tUAU)\ ).$$

Def. Le problème de **l'équilibrage d'une matrice** consiste à chercher à chercher deux matrices D_1 et D_2 diagonales inversibles telle que $\mathrm{cond}(D_1AD_2) = \inf_{\Delta_1,\Delta_2 \text{ diagonales inversibles}} \mathrm{cond}(\Delta_1A\Delta_2)$.

On résout alors $Ax = b$ en deux étapes : résolution de $D_1AD_2y = D_1b$ puis de $x = D_2y$.

En pratique on essaie plus simplement de minimiser le rapport entre le plus grand et le plus petit élément non nul de $A' = \Delta_1A\Delta_2$. Posons $E = \{(i,j) \in [\![1\,;n]\!]^2 \mid a'_{i,j} \neq 0\}$. On minimise $\frac{\max_{(i,j)\in E}|a'_{i,j}|}{\min_{(i,j)\in E}|a'_{i,j}|}$.

1.2 Conditionnement d'un problème de recherche de valeurs propres

Th. *Soit A diagonalisable et P une matrice telle que $P^{-1}AP = \mathrm{diag}(\lambda_i)$. Soit $\|\cdot\|$ une norme matricielle telle que, pour toute matrice diagonale* $\mathrm{diag}(\alpha_i)$ *: $\|\mathrm{diag}(\alpha_i)\| = \max_i|\alpha_i|$. Alors, pour toute matrice δA,* $\mathrm{Sp}(A+\delta A) \subset \bigcup_{i=1}^n D_i$ *avec* $D_i = \{z \in \mathbf{C} \mid |z-\lambda_i| \leqslant \mathrm{cond}_{\|\cdot\|}(P)\cdot\|\delta A\|\}$.

Def. Le conditionnement $\Gamma(A)$ relatif à la recherche des valeurs propres est le minimum de $\mathrm{cond}_{\|\cdot\|}(P)$ pour P tel que $P^{-1}AP$ soit diagonale.

On a alors $\mathrm{Sp}(A+\delta A) \subset \bigcup_{i=1}^n B(\lambda_i, \Gamma(A)\cdot\|\delta A\|)$.

Th. *Soit A symétrique et $B = A + \delta A$ où la perturbation δA est également symétrique. Soit $\alpha_1 \leqslant \ldots \leqslant \alpha_n$ les valeurs propres de A et $\beta_1 \leqslant \ldots \leqslant \beta_n$ celles de B. Alors $\forall i \in [\![1\,;n]\!], |\alpha_i - \beta_i| \leqslant \|\delta A\|_2$.*

2 Résolution de systèmes linéaires

Problème : résoudre $Ax = b$ sachant A inversible.

2.1 Méthode de Gauss (pour une matrice quelconque)

Par combinaisons linéaires successives entre lignes de A on se ramène à $(MA)x = Mb$ où MA est triangulaire supérieure. On résout ensuite cette équation par une méthode de remontée.

Un pivot trop petit en valeur absolue peut causer des erreurs d'arrondi du fait de la division par le pivot. D'où deux stratégies :

Algorithme 1 : Étape d'élimination

Entrées : Matrice A.
$A' \leftarrow A$;
tant que $\dim(A') > 1$ **faire**
  `pivot` $\leftarrow a'_{i,1} \neq 0$ (qui existe car A' est inversible) ;
  si $i \neq 1$ **alors**
    $L_i \leftrightarrow L_1$;
  pour tous les $i > 1$ *tel que* $a_{i,1} \neq 0$ **faire**
    $L_i \leftarrow L_i - \frac{a_{i,1}}{a_{1,1}} L_1$;
    `// on annule tous les termes de la colonne du pivot situés sous la diagonale`
  `// notre matrice` A' `n'a alors que des 0 sous le premier terme, qui est non nul`
  `// on met alors de côté la première ligne et la première colonne de` A'
  $A' \leftarrow (a'_{i,j})_{2 \leqslant i,j}$
On replace les lignes et colonnes supprimées au fur et à mesure pour obtenir une matrice triangulaire (MA).

- *pivot partiel* : on prend le terme de plus grande valeur absolue de la colonne courante, sur ou en dessous de la diagonale,
- *pivot total* : on choisit le terme de plus grande valeur absolue de la matrice résiduelle, i.e. si on est à l'étape $n-k+1$, la matrice constituée des k dernières lignes et colonnes (plus couteux).

La complexité est en $O\left(\frac{2n^3}{3}\right)$ sans choix du pivot.

2.2 Méthode de Gauss-Jordan (variante de la précédente)

Dans la phase d'éliminations on élimine également les terme au-dessus de la diagonale. On obtient ainsi une matrice diagonale, ce qui permet de calculer efficacement l'inverse.

2.3 Factorisation LU

Th. *Soit* $A = (a_{i,j}) \in \mathcal{GL}_n(\mathbf{K})$ *telle que* $\forall k \in [\![1\,;n]\!], \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \in \mathcal{GL}_k(\mathbf{K})$. *Alors* A *admet une factorisation sous la forme* $A = LU$ *avec* L *triangulaire inférieure et* U *triangulaire supérieure. De plus on peut choisir* $\forall i \in [\![1\,;n]\!], (L)_{ii} = 1$ *et la décomposition est alors unique.*

Cela signifie que, dans l'algorithme de Gauss, les pivots successifs peuvent toujours être pris sur la diagonale. Si la factorisation échoue ou peut toujours permuter des lignes de A pour arriver à une matrice A' qui admet une factorisation LU.

Cette factorisation est utile lorsque plusieurs systèmes linéaires sont à résoudre : on résout un système sur L puis un système sur U.

2.4 Méthode de Cholesky (matrices symétriques définies positives)

Th. *Soit* $A \in \mathcal{S}_n^{++}(\mathbf{K})$. *Il existe une matrice triangulaire* B *vérifiant* $A = B^t B$. *De plus on peut imposer que tous les éléments diagonaux de* B *soient tous strictement positifs et la factorisation* $A = B^t B$ *est alors unique.*

En pratique, on calcule B colonne par colonne à partir de $\forall 1 \leqslant i \leqslant j \leqslant n, a_{ij} = \sum_{k=1}^{i} b_{ik} b_{jk} = a_{ji}$.

Rem. Le déterminant de la matrice peut alors se calculer facilement : $\det(A) = (b_{11} b_{22} \cdots b_{nn})^2$.

Un système $Ax = b$ devient alors $B^t B x = b$. Pour résoudre le système on résout $By = b$ puis ${}^t B x = y$. La complexité de la factorisation suivie de la résolution est en $O\left(\frac{n^3}{3}\right)$.

3 Valeurs propres et vecteurs propres : méthode de Jacobi

Def. Soit le polynôme $P(\lambda) = \lambda^n + a_1\lambda^{n-1} + \ldots + a_{n-1}\lambda + a_n$. Est dite « compagne du polynôme P » la matrice suivante :

$$C(P) = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \ldots & \ldots & -a_{n-1} & -a_n \\ 1 & 0 & & & & & \\ 0 & 1 & 0 & & & & \\ & 0 & 1 & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 0 & 1 & 0 & \\ & & & & 0 & 1 & 0 \end{pmatrix}$$

Prop. *Le polynôme caractéristique de $C(P)$ vaut $(-1)^n P(\lambda)$. La matrice a donc pour valeurs propres les racines de P.*

Ce lien prouve, par le théorème d'Abel, que la recherche des valeurs propres d'une matrice ne peut se faire en un nombre fini d'opérations au-delà de la dimension 5.

Méthode de Jacobi Soit $A \in \mathcal{S}_n^+(\mathbf{R})$ non diagonale. On dispose de p et q, $p < q$ tels que $a_{pq} \neq 0$. On définit la matrice suivante :

$$\Omega = I_n + \sin(\theta) \cdot (E_{pq} - E_{qp}) + (1 - \cos(\theta)) \cdot (E_{pp} + E_{qq}) \, .$$

C'est la matrice de rotation d'angle $-\theta$ dans le plan défini par les p^{e} et q^{e} vecteurs de la base (donc orthogonale). On pose $B = {}^t\Omega A \Omega \in \mathcal{S}_n(\mathbf{R})$, $c = \cos(\theta)$, $s = \sin(\theta)$ et $t = \tan(\theta)$. On a alors

$$\begin{cases} b_{ij} = b_{ji} = a_{ij} & \text{si } i \notin \{p,q\} \text{ et } j \notin \{p,q\} \\ b_{pi} = b_{ip} = c \cdot a_{pi} - s \cdot a_{qi} & \text{si } i \notin \{p,q\} \\ b_{qi} = b_{iq} = s \cdot a_{pi} + c \cdot a_{qi} & \text{si } i \notin \{p,q\} \\ b_{pp} = a_{pp} - t \cdot a_{pq} & \\ b_{qq} = a_{qq} + t \cdot a_{pq} & \end{cases} .$$

Th. *Soit $A \in \mathcal{S}_n^+(\mathbf{R})$ et B la matrice obtenue par la construction précédente. On a alors les relations*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^2 \qquad \sum_{i=1}^{n} a_{ii}^2 + 2a_{pq}^2 = \sum_{i=1}^{n} b_{ii}^2$$

(le poids de la matrice se déporte sur la diagonale).

Th. *La suite des matrices obtenues par la méthode de Jacobi est convergente et converge vers une matrice diagonale contenant les valeurs propres de A.*

Pour accélérer la convergence il vaut mieux prendre $|a_{pq}|$ maximum.

Th. *Si toutes les valeurs propres de A sont distinctes, alors la suite des produits des matrices Ω converge vers une matrice orthogonale dont les vecteurs colonnes constituent unn ensemble orthnormal de vecteurs propres de A.*

4 Programmation linéaire : l'algorithme du simplexe

La forme standard d'un problème d'optimisation est la suivante :

- on maximise une forme linéaire f de n variables $x_1, \ldots, x_n$,
- avec m contraintes linéaires : $\forall i \in [\![1\,;m]\!], \sum_{j=1}^{n} a_{ij}x_j \leqslant b$,
- et n contraintes de positivité : $\forall j \in [\![1\,;n]\!], x_j \geqslant 0$.

Ces équations déterminent un **polyèdre des contraintes** qui est convexe. Les n-uplets $(x_1, \ldots, x_n)$ qui satisfont ces contraintes s'appellent **solutions réalisables** du problème. Ce sont les points intérieurs (au sens large) du polyèdre des contraintes.

Th. *Soit un problème de programmation linéaire dont le polyèdre des contraintes est non vide et dont la fonction à maximiser est majorée sur ce polyèdre. Alors le problème admet un maximum fini atteint en au moins un sommet du polyèdre est contraintes.*

Pour trouver le maximum on passe alors itérativement d'un sommet à un sommet adjacent de façon à augmenter la valeur de f.