



TECHNIQUES
DE L'INGÉNIEUR

Réf. : AF485 V1

Méthodes numériques en algèbre linéaire

Date de publication :
10 octobre 1998

Cet article est issu de : **Sciences fondamentales | Mathématiques**

par **Robert CABANE**

Pour toute question :
Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Document téléchargé le : **06/04/2019**
Pour le compte : **7200043660 - centralesupelec // 195.221.160.3**

Par mail :
infos.clients@teching.com
Par téléphone :
00 33 [0]1 53 35 20 20

© Techniques de l'Ingénieur | tous droits réservés

Méthodes numériques en algèbre linéaire

par **Robert CABANE**

Ancien élève de l'École Normale Supérieure
Professeur de Mathématiques Spéciales
au Lycée Michel-Montaigne (Bordeaux)

1. Traitement des erreurs en algèbre linéaire.....	AF 485 - 2
1.1 Position du problème.....	— 2
1.2 Normes vectorielles, normes matricielles	— 2
1.3 Normes et rayon spectral.....	— 6
1.4 Conditionnement.....	— 8
1.5 Étude des erreurs et de leur propagation.....	— 9
2. Méthodes du pivot.....	— 11
2.1 Principe de l'échelonnement	— 11
2.2 Une variante : la méthode de Crout (dite LU)	— 14
2.3 Application à la résolution des systèmes linéaires.....	— 15
2.4 Calcul de déterminants	— 15
2.5 Obtention de la matrice inverse par la méthode de Gauss-Jordan	— 16
2.6 Recherche de relations de dépendance.....	— 16
2.7 Faut-il faire confiance à la méthode du pivot.....	— 16
3. Méthodes itératives.....	— 16
3.1 Principes théoriques.....	— 16
3.2 Méthode de Gauss-Jacobi.....	— 17
3.3 Méthode de Gauss-Seidel.....	— 18
3.4 Généralisation.....	— 18
3.5 Amélioration itérative des solutions.....	— 18
4. Méthodes euclidiennes.....	— 18
4.1 Un peu de théorie	— 19
4.2 Procédés d'orthogonalisation.....	— 20
4.3 Méthode de Cholesky.....	— 22
4.4 Décomposition en valeurs singulières.....	— 23
4.5 Pseudo-inverses.....	— 23
5. Matrices creuses	— 25
5.1 Problèmes de grande taille en algèbre linéaire	— 25
5.2 Modes de représentation d'une matrice creuse	— 25
5.3 Algorithmes spécifiques pour les matrices creuses	— 25
Pour en savoir plus.....	Doc. AF 485

Autant l'algèbre linéaire s'occupe de vecteurs très généraux, autant l'analyse numérique linéaire considère essentiellement des vecteurs ayant un nombre fini de composantes numériques, c'est-à-dire situés dans des espaces de dimension finie. Le but de cet ensemble de méthodes est de dégager des procédures explicites qui conduisent à des approximations aussi précises que possible des objets « idéaux » que la théorie a dégagés.

On verra assez rapidement que la notion de précision est elle-même imprécise, car on peut accepter, ou non, une certaine marge d'erreur sur les résultats, et mesurer cette erreur par divers procédés. Nous chercherons donc à dégager en quel(s) sens un vecteur peut être considéré comme « petit », une solution « acceptable ». L'étude rigoureuse des erreurs et de leur propagation au cours des calculs est cependant difficile et amène généralement des résultats exagérément pessimistes. Des points de vue différents, fondés sur la théorie des probabilités, conduisent souvent à des conclusions plus engageantes.

Cette étude, poussée à son extrême limite, nous amènera à une impasse dans la mesure où certains concepts de l'algèbre linéaire s'exprime par des valeurs entières (ce sont des dimensions), pour lesquelles la notion de valeur approchée n'a aucun sens.

La notion d'algorithme apparaîtra vite prépondérante ; en effet, c'est par une itération que l'on parvient généralement à « calculer » les objets recherchés. Pour prendre un exemple très simple, le produit scalaire de deux vecteurs v et w ayant n composantes se calcule par l'algorithme suivant :

Initialiser une somme S à 0.

Faire varier un compteur i de 1 à n .

Pour chaque valeur de i , ajouter $v_i w_i$ à S .

Le résultat est la valeur finale de S .

Nous présenterons les algorithmes « en français », sans faire référence à un langage informatique particulier. De fait, la plupart sinon la totalité des algorithmes signalés se trouvent déjà codés dans l'une des bibliothèques de programmes existantes, en Fortran ou en C. Il n'est pas très difficile d'adapter ces mêmes algorithmes à d'autres langages de programmation.

Enfin, ce domaine aux confins de l'Algèbre et de l'Analyse a connu un certain renouvellement sous l'influence grandissante des logiciels qui permettent un calcul formel, c'est-à-dire exact et non approché. Ces produits, bien au point depuis les années 1990, permettent d'aborder plus favorablement la recherche des grandeurs entières dont on a parlé plus haut. Dans ces conditions, se pose la question du calcul effectif de certains objets de l'Algèbre linéaire comme les vecteurs propres ; ainsi, le travail « formel » sur les valeurs propres conduit tout naturellement à calculer dans des corps de nombres algébriques.

Nous invitons le lecteur à se reporter à l'article général [AF 85] Algèbre linéaire pour les bases et les notations les plus courantes de cette théorie ; il pourra également consulter l'article relatif aux structures euclidiennes.

Le présent article se limite aux méthodes de résolution exacte ou approchée des équations linéaires (vectorielles), et aux outils théoriques relatifs à ces méthodes. Les problèmes de calcul exact ou approché des éléments propres (valeurs propres, vecteurs propres) seront traités dans un autre article.

1. Traitement des erreurs en algèbre linéaire

1.1 Position du problème

Le problème posé est essentiellement celui-ci : étant donné un opérateur linéaire u , si un vecteur x est connu à une précision ε , quelle précision peut-on espérer quant au vecteur $u(x)$? La notion de précision d'un vecteur sera détaillée par l'introduction de normes. On étudiera aussi le problème inverse, relatif à u^{-1} , qui est

essentiellement celui de la détermination de la précision d'une solution d'une équation linéaire.

Le lecteur pourra consulter avec profit l'article [A 101] Analyse fonctionnelle.

1.2 Normes vectorielles, normes matricielles

On se propose ici de définir la notion de « grandeur » d'un vecteur ou d'un opérateur linéaire.

1.2.1 Notations

Dans cet article nous considérerons des espaces vectoriels basés sur le corps des nombres réels ou sur le corps des nombres complexes. Pour simplifier, lorsque la distinction ne sera pas essentielle, nous noterons \mathbb{K} ce corps. Les espaces vectoriels seront notés E , $F\dots$ et les vecteurs seront notés $x, y\dots$

Les applications linéaires seront notées $u, v\dots$

Les normes sur les espaces vectoriels seront généralement notées $N, N', N_1, N_2\dots$ avec une exception notable : les normes d'applications linéaires (ou normes matricielles) seront plutôt notées $\|\cdot\|$.

L'ensemble des matrices ayant n lignes et p colonnes, à coefficients dans le corps \mathbb{K} , noté $\mathfrak{M}_{n,p}(\mathbb{K})$; lorsque ces matrices sont carrées ($n = p$), on note plus simplement $\mathfrak{M}_n(\mathbb{K})$ (c'est une algèbre identifiable à $\mathcal{L}(\mathbb{K}^n)$). En conséquence, nous identifierons un vecteur x de \mathbb{K}^n avec le n -uplet de ses coordonnées

(x_1, \dots, x_n) ainsi qu'avec la matrice unicolonne :

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

La matrice-identité de ce dernier espace sera noté I_n .

Etant donnée une matrice carrée $A = (a_{ij})$ à coefficients réels ou complexes, on appelle **adjointe** de A la matrice $A^* = (b_{ij})$ telle que $b_{ij} = \overline{a_{ji}}$. Dans le cas où A est une matrice réelle, B s'appelle aussi la matrice transposée de A .

1.2.2 Définitions

Définitions 1. Soit un espace vectoriel E .

On appelle norme sur E toute application N de E dans \mathbb{R}_+ satisfaisant les trois axiomes suivants :

(N1) $N(0) = 0$ et pour tout vecteur $x \in E$, $N(x) = 0$ entraîne $x = 0$;

(N2) $N(\lambda \cdot x) = |\lambda| N(x)$ pour tout vecteur $x \in E$ et tout scalaire λ (égalité d'homogénéité) ;

(N3) $N(x + y) \leq N(x) + N(y)$ pour tous vecteurs $x, y \in E$ (inégalité triangulaire).

On dit qu'un vecteur x de E est **unitaire** si sa norme vaut 1. À tout vecteur x non nul on peut associer un vecteur unitaire qui est $\frac{x}{N(x)}$.

La **distance associée** à la norme N est l'application qui, à un couple de vecteurs (x, y) , associe :

$$d(x, y) = N(x - y)$$

Elle vérifie immédiatement les trois axiomes suivants :

(D1) $d(x, y) = 0$ équivaut à $x = y$;

(D2) $d(x, y) = d(y, x)$ pour tous vecteurs $x, y \in E$ (égalité de symétrie) ;

(D3) $d(x, y) \leq d(x, z) + d(z, y)$ pour tous vecteurs $x, y, z \in E$ (inégalité triangulaire).

On note souvent $\|x\|$ au lieu de $N(x)$, s'inspirant de la valeur absolue. On rappelle (voir l'article A 101 *Analyse fonctionnelle*) que, sur un espace vectoriel de dimension finie, toutes les normes sont

équivalentes, c'est-à-dire que si N_1 et N_2 sont deux normes sur E , il existe deux constantes a et b strictement positives, telles que :

$$aN_1 \leq N_2 \leq bN_1$$

Définition 2. On appelle **algèbre normée** toute \mathbb{K} -algèbre $(E, +, \cdot, *)$ munie d'une norme N telle que :

$$N(x * y) \leq N(x)N(y)$$

pour tous éléments x, y de E .

Si la loi $*$ possède un élément unité e , la structure $(E, +, *)$ étant alors une algèbre unitaire, on a nécessairement :

$$N(e) \leq N(e)^2, \text{ c'est-à-dire } N(e) \geq 1$$

Si, dans ces conditions, un élément x possède un inverse x^{-1} , alors on a :

$$1 \leq N(e) = N(x \cdot x^{-1}) \leq N(x)N(x^{-1}),$$

soit encore :

$$N(x^{-1}) \geq \frac{N(e)}{N(x)} \geq \frac{1}{N(x)}$$

Nous reviendrons sur cette question à propos du conditionnement, au paragraphe 1.4.

Considérons à présent deux espaces vectoriels E et F munis de normes N et N' ; en pratique, on a le plus souvent :

$$F = \mathbb{K}^n \quad \text{et} \quad E = \mathbb{K}^p$$

de sorte que les applications linéaires de E dans F ne sont autres que des matrices à n lignes et p colonnes. La plupart des énoncés qui suivent s'expriment indifféremment dans le langage des applications linéaires et dans celui des matrices.

Définition 3. On note $\mathcal{L}(E, F)$ l'espace vectoriel des applications linéaires continues de E dans F .

Lorsque $E = F$, on note plus particulièrement :

$$\mathcal{L}(E) = \mathcal{L}(E, E)$$

l'algèbre des endomorphismes continus d'un espace vectoriel normé E .

On rappelle qu'une application linéaire u est continue si, et seulement si, elle est lipschitzienne, c'est-à-dire qu'il existe une constante k réelle telle que l'on ait $|u(x)| \leq kN(x)$ pour tout vecteur $x \in E$. En dimension finie, cette condition est toujours réalisée, de sorte que l'espace des applications linéaires de E dans F coïncide avec $\mathcal{L}(E, F)$.

1.2.3 Normes subordonnées

Définition 4. Dans le cadre précédent, on appelle **norme subordonnée** à N et N' l'application de $\mathcal{L}(E, F)$ dans \mathbb{R} qui, à une application linéaire continue u , associe :

$$\|u\| = \sup_{N(x) \leq 1} N'(u(x))$$

L'article A 101 *Analyse fonctionnelle* désigne cette norme particulière par l'appellation « triple norme », avec la notation $\||u|\|$.

Proposition 1.

La norme subordonnée d'une application linéaire continue u existe effectivement ; elle fournit une norme sur $\mathcal{L}(E, F)$; enfin, elle est telle que, pour tout vecteur x , on ait :

$$N'(u(x)) \leq \|u\| N(x)$$

Proposition 2.

Toute norme subordonnée est une norme d'algèbre sur $\mathcal{L}(E)$. Plus généralement, si u et v sont des applications linéaires continues entre des espaces normés telles que $v \circ u$ existe, alors $v \circ u$ est continue et $\|v \circ u\| \leq \|v\| \|u\|$, les normes utilisées étant toutes trois subordonnées aux normes des espaces sous-jacents.

Il ne faut pas croire que toute norme sur $\mathcal{L}(E, F)$ soit nécessairement subordonnée à des normes convenables sur E et F . Ainsi, toute norme $\|\cdot\|$ telle que $\|Id\| < 1$ ne peut être une norme d'algèbre, donc pas non plus une norme subordonnée. Et une norme telle que $\|Id\| > 1$ ne peut pas non plus être une norme subordonnée.

Voici un exemple de norme d'algèbre non subordonnée, la **norme de Schur**. Étant donnée une matrice $A = (a_{ij})$, carrée de taille n à coefficients réels ou complexes, on pose :

$$\|A\|_s = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\text{tr} A^* A}$$

On vérifie aisément qu'il s'agit d'une norme sur $\mathfrak{M}_n(\mathbb{K})$: c'est en fait la norme euclidienne (ou hermitienne) usuelle de l'espace \mathbb{K}^{n^2} . Comme $\|I_n\|_s = \sqrt{n}$, on voit que cette norme n'est pas subordonnée. Cependant, c'est une norme d'algèbre car :

$$\begin{aligned} \|AB\|_s^2 &= \sum_{i,j} \left| \sum_k a_{ik} b_{kj} \right|^2 \\ &\leq \left(\sum_{i,j} \left| a_{ik} \right|^2 \right) \left(\sum_{k,l} \left| b_{kj} \right|^2 \right) \text{ (inégalité de Cauchy-Schwarz)} \\ &= \sum_{i,j,k,\ell} |a_{ik}|^2 |b_{kj}|^2 = \|A\|_s^2 \|B\|_s^2 \end{aligned}$$

Cette norme est souvent utilisée à la place de la norme subordonnée à la norme euclidienne, dont on parlera au paragraphe 1.3.3.

1.2.4 Exemples de normes et normes subordonnées

Sur l'espace de dimension finie \mathbb{K}^n (auquel tout espace de dimension finie se ramène par le choix d'une base), on va définir aisément **trois normes** qui seront importantes pour la suite. Soit un vecteur $x = (x_1, \dots, x_n) \in \mathbb{K}^n$.

N_1 est définie par :

$$N_1(x) = |x_1| + \dots + |x_n|$$

N_2 est définie par :

$$N_2(x) = \sqrt{|x_1|^2 + \dots + |x_n|^2}$$

on dit qu'il s'agit d'une norme euclidienne (réelle ou complexe)

parce qu'elle est associée au produit scalaire $(x | y) = \sum_{i=1}^n \bar{x}_i y_i$, sur

l'espace $E = \mathbb{K}^n$, par la formule $N_2(x) = \sqrt{(x | x)}$.

N_∞ est définie par :

$$N_\infty(x) = \max_{1 \leq i \leq n} |x_i|$$

Calculons à présent la **norme subordonnée (matricielle)** associée à la norme N_∞ sur \mathbb{K}^n . Soit u l'endomorphisme de \mathbb{K}^n associé à la matrice $A = (a_{ij})$.

D'une part, on a pour $N_\infty(x) \leq 1$:

$$N_\infty(Ax) = \max_i \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \max_i \sum_{j=1}^n |a_{ij}| |x_j| \leq 1 \cdot \max_i \sum_{j=1}^n |a_{ij}|$$

$$\text{donc } \|A\| \leq \max_i \sum_{j=1}^n |a_{ij}|$$

D'autre part, choisissons $i = i_0$ tel que :

$$\max_i \sum_{j=1}^n |a_{ij}| = \sum_{j=1}^n |a_{i_0 j}|$$

et considérons $x_j = 1$ si $a_{i_0 j} = 0$ et $x_j = \frac{a_{i_0 j}}{|a_{i_0 j}|}$ sinon ; de cette façon, $a_{i_0 j} x_j = |a_{i_0 j}|$ dans tous les cas.

Ainsi, on a bien $N_\infty(x) = 1$ et la composante de numéro i_0 de Ax s'écrit :

$$\sum_{j=1}^n a_{i_0 j} x_j = \sum_{j=1}^n |a_{i_0 j}| = \max_i \sum_{j=1}^n |a_{ij}|$$

donc $\|Ax\| \geq \max_i \sum_{j=1}^n |a_{ij}|$, ce qui entraîne l'égalité. En fin de compte, on a :

$$\|A\| = \max_i \sum_{j=1}^n |a_{ij}|$$

On peut donc poser la définition 5.

Définition 5. On appelle **norme des lignes** la norme définie sur $\mathfrak{M}_n(\mathbb{K})$ par :

$$\|A\|_\ell = \max_i \sum_{j=1}^n |a_{ij}|$$

C'est la norme subordonnée à la norme N_∞ sur l'espace \mathbb{K}^n .

On a de même la définition 6.

Définition 6. On appelle **norme des colonnes** la norme définie sur $\mathfrak{M}_n(\mathbb{K})$ par :

$$\|A\|_c = \max_j \sum_{i=1}^n |a_{ij}|$$

C'est la norme subordonnée à la norme N_1 sur l'espace \mathbb{K}^n .

Enfin, nous introduisons une norme très importante au plan théorique.

Définition 7. On appelle **2-norme** sur $\mathbb{M}_n(\mathbb{K})$ la norme subordonnée à la norme N_2 sur l'espace \mathbb{K}^n .

Cette norme sera notée $\|\cdot\|_2$.

Le calcul effectif de cette 2-norme est délicat, comme on le verra plus loin au paragraphe 1.3.3.

1.2.5 Normes compatibles

Les exemples précédents laissent penser qu'il n'est pas toujours facile de calculer explicitement une norme subordonnée ; d'autre part, le choix d'une norme est souvent un peu arbitraire.

Définition 8. Soient une norme N sur \mathbb{K}^n et une norme d'algèbre $\|\cdot\|$ sur $\mathbb{M}_n(\mathbb{K})$. On dit que la seconde est **compatible** avec la première si l'on a, pour tout vecteur v et matrice A , l'inégalité :

$$N(Av) \leq \|A\| N(v)$$

Ce qui revient à avoir :

$$\|A\|_N \leq \|A\|$$

$\|\cdot\|_N$ désignant la norme subordonnée à N .

Par exemple, considérons la norme de Schur $\|\cdot\|_s$ introduite précédemment au § 1.2.3). Elle est compatible avec la norme $\|\cdot\|_2$. En effet, étant donné un vecteur v de composantes v_j , on a (grâce à l'inégalité de Cauchy-Schwarz) :

$$N_2(Av)^2 = \sum_i \left| \left(\sum_j a_{ij} v_j \right) \right|^2 \leq \sum_i \left(\sum_j |a_{ij}|^2 \sum_j |v_j|^2 \right) = \|A\|_s^2 N_2(v)^2$$

Proposition 3.

Toute norme d'algèbre sur $\mathbb{M}_n(\mathbb{K})$ est compatible avec au moins une norme vectorielle sur \mathbb{K}^n .

Preuve. ◊ Soit $\|\cdot\|$ une norme d'algèbre sur $\mathbb{M}_n(\mathbb{K})$. Soit un vecteur V de \mathbb{K}^n . Introduisons la matrice \tilde{V} dont la première colonne est V et les colonnes suivantes sont nulles. Posons :

$$N(V) = \|\tilde{V}\|$$

on vérifie facilement que c'est une norme sur \mathbb{K}^n . De plus, $\tilde{A}\tilde{V}$ est une matrice ayant pour première colonne AV , suivie de colonnes nulles ; par conséquent, on a :

$$N(AV) = \|\tilde{A}\tilde{V}\| = \|A\tilde{V}\| \leq \|A\| \|\tilde{V}\| = \|A\| N(V)$$

ce qu'il fallait. ◊

1.2.6 Application à l'évaluation des erreurs

Considérons une matrice $A \in \mathbb{M}_n(\mathbb{K})$, une norme N sur \mathbb{C}^n et la norme subordonnée $\|\cdot\|$ sur $\mathbb{M}_n(\mathbb{K})$. Dans de nombreux problèmes pratiques, on doit calculer l'image d'un vecteur x non nul par A . Cependant, le vecteur x n'est peut-être connu qu'à une certaine pré-

cision, de sorte qu'on calcule en réalité avec un vecteur $x + \Delta x$ (nous supposons que la matrice A est connue sans erreur). La grandeur de l'erreur est donnée par $N(\Delta x)$. L'image (fausse) obtenue est $Ax + A\Delta x$ et l'erreur sur l'image est $A\Delta x$; la grandeur de cette erreur est :

$$N(A\Delta x) \leq \|A\| N(\Delta x)$$

Dans le cas où on applique A de manière itérative (par exemple, dans un schéma d'approximations successives ou une chaîne de Markov), on cherche à approcher $A^p x$. Posons :

$$x_0 = x + \Delta x'; x_1 = Ax_0 = Ax + A\Delta x, \text{ et ainsi de suite.}$$

On a donc :

$$A^p x_0 = x_p = A^p x + A^p \Delta x$$

il vient une majoration de l'erreur (absolue) commise sur $A^p x$ de la forme :

$$N(A^p \Delta x) \leq \|A^p\| N(\Delta x)$$

En général, on doit donc craindre une croissance géométrique des erreurs, sauf si $\|A\| \leq 1$. Notons que $\|A^p\|$ ne croît pas nécessairement de manière géométrique, particulièrement si A n'est pas diagonalisable ; par exemple, on a :

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^p = \begin{pmatrix} 1 & p \\ 0 & 1 \end{pmatrix}$$

qui a une croissance arithmétique.

Terminons par un exemple. Soit la matrice :

$$A = \begin{pmatrix} 0 & 100 \\ 0 & 0 \end{pmatrix}$$

qui est singulière, de norme 100 (en lignes ou colonnes).

Le vecteur $X_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ est tel que $AX_0 = 0$;

le vecteur $X = \begin{pmatrix} 1 \\ 0,01 \end{pmatrix}$ est tel que $AX = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Ici, ΔX est de l'ordre de grandeur de 10^{-2} tandis que $AX - AX_0 = A\Delta X$ est de l'ordre de grandeur de 1. Cependant, le vecteur $X' = \begin{pmatrix} 1,01 \\ 0,01 \end{pmatrix}$ vérifie $AX' = AX$, avec une erreur nulle.

Ainsi, l'erreur commise sur l'image dépend fortement de la direction suivie.

Si on considère les erreurs relatives, on peut poser $\frac{N(\Delta X)}{N(X)} = \varepsilon$. On voit que l'**erreur relative sur l'image** s'écrit :

$$\frac{N(A\Delta X)}{N(Ax + A\Delta X)} \leq \frac{\|A\| N(\Delta X)}{N(Ax) - \|A\| N(\Delta X)}$$

Comme A est inversible et que l'on utilise une norme subordonnée, on trouve la majoration :

$$N(X) = N(A^{-1}Y) \leq \|A^{-1}\| N(Y) = \|A^{-1}\| N(AX)$$

donc :

$$\frac{\|A\| N(\Delta X)}{N(AX) - \|A\| N(\Delta X)} \leq \frac{\text{cond}(A) N(\Delta X)}{N(X) - \text{cond}(A) N(\Delta X)} = \frac{\text{cond}(A)\varepsilon}{1 - \text{cond}(A)\varepsilon}$$

où l'on a utilisé le **conditionnement** de A , défini comme :

$$\text{cond}(A) = \|A\| \|A^{-1}\|$$

On voit que, en gros, si l'erreur relative sur X est suffisamment petite, alors l'erreur relative sur l'image est au pire de l'ordre de grandeur de l'erreur relative sur l'antécédent multipliée par le conditionnement de la matrice.

1.3 Normes et rayon spectral

1.3.1 Définition et comparaison

Dans ce paragraphe, nous nous occuperons que de matrices carrées complexes.

Définition 9. Soit une matrice $A \in \mathfrak{M}_n(\mathbb{C})$. On appelle **rayon spectral** de A le nombre :

$$\rho(A) = \max_{\lambda \in \text{Sp}(A)} |\lambda|$$

notant $\text{Sp}(A)$ le spectre de A (ensemble des valeurs propres). C'est donc le plus grand module possible d'une valeur propre de A .

Proposition 4.

Pour toute matrice A et toute norme d'algèbre $\|\cdot\|$, on a :

$$\rho(A) \leq \|A\|$$

Preuve. ◇ Choisissons une norme vectorielle N , compatible avec la norme proposée (on a vu que c'est possible). Si λ est la valeur propre de A de plus grand module, associée à v , alors :

$$N(Av) = |\lambda| N(v) \leq \|A\| N(v)$$

par définition de la norme compatible ; le résultat en découle. ◇

Corollaire 1.

On a, pour toute norme d'algèbre $\|\cdot\|$, pour toute matrice A et tout entier k , la majoration :

$$\rho(A) \leq \sqrt[k]{\|A^k\|}$$

Preuve. ◇ C'est simplement que $\rho(A^k) \leq \|A^k\|$ et que par permanence des valeurs propres :

$$\rho(A)^k = \rho(A^k) \quad \diamond$$

En fait, cette inégalité devient d'autant plus favorable que k est grand.

Théorème 1.

On a, pour toute matrice A :

$$\rho(A) = \lim_{k \rightarrow \infty} \sqrt[k]{\|A^k\|}$$

Preuve. ◇ Changeons d'abord de base pour amener A à une forme de Jordan, et choisissons comme norme la norme des lignes associée à la nouvelle base de \mathbb{C}^n . La norme de A^k est simplement la plus grande des normes des puissances des blocs de Jordan de A . On peut ignorer les blocs de Jordan associés à une valeur propre nulle, parce que ces blocs sont nilpotents et n'influent pas sur le calcul de $\|A^k\|$ quand k est grand.

On considère donc un bloc de Jordan, de la forme $J = \lambda I_p + N$, N étant une matrice nilpotente d'indice p (telle que $N^p = 0$ et $N^{p-1} \neq 0$). Il vient pour $k > p$:

$$J^k = \lambda^k + \sum_{j=1}^{p-1} C_k^j \lambda^{k-j} N^j$$

Dans cette somme, le dernier terme est prépondérant sur les autres lorsque k tend vers l'infini à cause du coefficient binomial et de la non-nullité de N^{p-1} . On a donc :

$$J^k = C_k^{p-1} \lambda^k (N^{p-1} + o(1))$$

le terme noté $o(1)$ désignant diverses matrices de limites nulles ; par conséquent il vient (avec une norme adéquate, obtenue par restriction) :

$$\|J^k\| \leq C_k^{p-1} |\lambda|^k (\|N^{p-1}\| + o(1)) \sim \frac{k^{p-1}}{(p-1)!} |\lambda|^k \|N^{p-1}\|$$

d'où :

$$\frac{1}{k} \ln \|J^k\| \leq (p-1) \frac{\ln k}{k} + \ln |\lambda| + o(1)$$

ce qui tend vers $\ln |\lambda|$. Faisant cela pour tous les blocs de Jordan de A , on voit que (pour la norme choisie) $\sqrt[k]{\|A^k\|}$ tend vers $\rho(A)$.

Pour une autre norme d'algèbre v sur $\mathfrak{M}_n(\mathbb{C})$, on a a priori équivalence donc l'existence d'une constante b telle que :

$$v \leq b \|\cdot\|$$

On en tire :

$$\rho(A) \leq \sqrt[p]{v(A^k)} \leq \sqrt[p]{b} \sqrt[k]{\|A^k\|}$$

et le majorant tend bien vers $\rho(A)$. ◇

1.3.2 Application aux études de convergence

Corollaire 2.

La suite de matrices (A^k) tend vers la matrice nulle si, et seulement si, le rayon spectral de A vérifie :

$$\rho(A) < 1$$

Preuve. ◇ Nous choisissons une norme $\|\cdot\|$, subordonnée à une norme N sur \mathbb{C}^n . Si $\rho(A) < 1$, alors $\lim_{k \rightarrow \infty} \frac{1}{k} \ln \|A^k\| < 0$, ce qui montre que $\ln \|A^k\|$ est équivalent à $k \ln \rho(A)$ et tend vers $-\infty$. Par conséquent, $\|A^k\|$ tend vers 0.

Réciproquement, si $\|A^k\|$ tend vers 0, d'après le corollaire 1 du paragraphe 1.3.1, $\rho(A)^k \leq \|A^k\|$ tend vers 0 aussi, ce qui impose que $\rho(A) < 1$. ◇

Corollaire 3.

La série géométrique matricielle $\sum_{k=0}^{\infty} A^k$ (avec $A^0 = I_n$) converge si, et seulement si, $\rho(A) < 1$; dans ce cas, sa somme est l'inverse de $I - A$.

Preuve. ◇ Pour que cette série converge, il est nécessaire que son terme général tende vers 0, donc que $\|A^p\|$ tende vers 0, ce qui entraîne $\rho(A) < 1$. Réciproquement, on peut écrire :

$$(I - A) \sum_{k=0}^p A^k = I - A^{p+1}$$

si $\rho(A) < 1$ alors A^{p+1} tend vers 0 ; de plus, 1 ne peut être valeur propre de A et la matrice $I - A$ est inversible ; on a donc

$$\sum_{k=0}^p A^k = (I - A)^{-1} (I - A^{p+1}) \text{ qui converge vers } (I - A)^{-1}. \diamond$$

1.3.3 Norme euclidienne

Nous pouvons, à présent, affiner l'étude de la 2-norme $\|\cdot\|_2$. Dans ce paragraphe, une norme euclidienne N est fixée, issue d'un produit scalaire noté $(\cdot | \cdot)$ sur l'espace $E = \mathbb{K}^n$, définissant la norme subordonnée associée $\|\cdot\|_2$.

Proposition 5.

Pour tout endomorphisme u d'un espace euclidien (réel ou complexe) on a :

$$\|u\|_2 = \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |(u(x)|y)|$$

Preuve. \diamond Soit S la borne supérieure figurant à droite. Elle est finie car, selon l'inégalité de Cauchy-Schwarz, on a :

$$|(u(x)|y)| \leq N(u(x)) N(y) \leq N(u(x)) \leq \|u\|_2 N(x) \leq \|u\|_2$$

On a donc aussitôt

$$S \leq \|u\|_2$$

D'autre part, soit $N(x) = 1$. Si $u(x) = 0$ alors :

$$N(u(x)) = 0 = |(u(x)|y)| \leq S$$

pour tout y .

Sinon, définissons $y = \frac{1}{N(u(x))} u(x)$, de sorte que $N(y) = 1$ et que :

$$(u(x)|y) = \frac{1}{N(u(x))} (u(x)|u(x)) = \frac{N(u(x))^2}{N(u(x))} = N(u(x))$$

donc à nouveau $N(u(x)) \leq S$.

En passant à la borne supérieure sur x , on obtient $\|u\|_2 \leq S$. \diamond

Proposition 6.

On a les relations :

$$\|u^*\|_2 = \|u\|_2 \text{ et } \|u\|_2^2 = \|u \circ u^*\|_2 = \|u^* \circ u\|_2$$

u^* désignant l'adjoint de l'endomorphisme u .

Preuve. \diamond D'après la proposition précédente :

$$\|u^*\|_2 = \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |(u^*(x)|y)| = \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |(u(y)|x)| = \|u\|_2$$

Comme la norme subordonnée est une norme d'algèbre, on a certainement :

$$\|u^* \circ u\|_2 \leq \|u\|_2 \|u^*\|_2 = \|u\|_2^2$$

Cependant, appliquant la même idée que ci-dessus (choisir un vecteur x de norme 1), on trouve que :

$$N(u(x))^2 = (u(x)|u(x)) \leq \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |(u(x)|u(y))|$$

$$= \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |u^* \circ u(x)|y| = \|u^* \circ u\|_2$$

d'où le résultat en passant à la borne supérieure. \diamond

Corollaire 4.

Pour tout endomorphisme u d'un espace euclidien et tout automorphisme v unitaire (c'est-à-dire tel que $v^{-1} = v^*$), on a :

$$\|u\|_2 = \|v \circ u\|_2 = \|u \circ v\|_2$$

Preuve. \diamond

$$(u \circ v) \circ (u \circ v)^* = u \circ v \circ v^* \circ u^* = u \circ u$$

et

$$(v \circ u)^* \circ (v \circ u) = u^* \circ v^* \circ v \circ u = u^* \circ u \diamond$$

Proposition 7.

Lorsque u est un endomorphisme autoadjoint positif (c'est-à-dire tel que $u = u^*$ et que $(u(x)|x) \geq 0$ pour tout x), on a :

$$\|u\|_2 = \sup_{N(x) \leq 1} (u(x)|x) = \rho(u)$$

Preuve. \diamond On sait que $\|u\|_2 = \sup_{\substack{N(x) \leq 1 \\ N(y) \leq 1}} |(u(x)|y)|$

Donc, si $S = \sup_{N(x) \leq 1} (u(x)|x)$ alors $S \leq \|u\|_2$

Soit (v_1, \dots, v_n) une base orthonormée de vecteurs propres de u . On a pour tout vecteur x de norme 1 :

$$x = \sum_{i=1}^n x_i v_i,$$

$$u(x) = \sum_{i=1}^n x_i \lambda_i v_i,$$

$$N(x)^2 = \sum x_i^2$$

et

$$N(u(x))^2 = \sum \lambda_i^2 x_i^2 \leq \rho(u)^2 N(x)^2 = \rho(u)^2$$

donc

$$\|u\|_2 \leq \rho(u)$$

Enfin, si λ_n est la plus grande valeur propre de u , associée à v_n , alors

$$(u(v_n)|v_n) = \lambda_n = \rho(u) \leq S$$

ce qui achève la preuve. ◇

Corollaire 5.

Pour tout endomorphisme u d'un espace euclidien ou hermitien E on a :

$$\|u\|_2 = \sqrt{\rho(u^* \circ u)}$$

On conçoit alors que le calcul effectif de la norme de u soit délicat : il requiert la recherche de la plus grande valeur propre d'un endomorphisme autoadjoint positif (c'est-à-dire d'une matrice symétrique ou hermitienne positive).

1.4 Conditionnement

1.4.1 Définition et calcul

Soit une norme d'algèbre $\|\cdot\|$ sur $\mathcal{L}(E)$. On appelle **conditionnement** d'une application linéaire inversible u le nombre :

$$\text{cond}(u) = \|u\| \|u^{-1}\|$$

Le conditionnement d'une matrice se définit de manière analogue.

On a certainement en raison de l'hypothèse de norme d'algèbre :

$$\text{cond}(u) \geq \|u \circ u^{-1}\| = \|\text{Id}\| \geq 1$$

Le conditionnement d'une matrice est invariant par adjonction, par passage à l'inverse et par multiplication par une constante.

Il est possible de calculer théoriquement le conditionnement lorsqu'on utilise la 2-norme sur $\mathcal{L}(E)$.

Proposition 8.

Soit $\text{cond}_2(A)$ le conditionnement d'une matrice A calculé à partie de la 2-norme $\|\cdot\|_2$.

Soient $(\lambda_1, \dots, \lambda_n)$ les valeurs propres de A^*A , rangées en ordre croissant.

Alors :

$$\text{cond}_2(A) = \sqrt{\frac{\lambda_n}{\lambda_1}}$$

Preuve. ◇ On a vu que :

$$\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = \sqrt{\lambda_n}$$

et l'égalité $A^{-1*}A^{-1} = (AA^*)^{-1}$ montrer que $\rho(A^{-1*}A^{-1}) = \lambda_1$, ce qui amène la formule. ◇

Les nombres $\sqrt{\lambda_i}$ s'appellent les **valeurs singulières** de A ; il en est discuté au paragraphe consacré à la décomposition en valeurs singulières (§ 4.4).

Nous voyons que, pour la 2-norme sur $\mathfrak{M}_n(\mathbb{K})$, le conditionnement d'une matrice A ne vaut 1 que si A^*A est scalaire, c'est-à-dire que A est proportionnelle à une matrice unitaire.

En pratique, le calcul du conditionnement d'une matrice est plus aisés si on se réfère aux normes vectorielles N_1 ou N_∞ , pour lesquelles la norme subordonnée se calcule relativement bien. Il faut, quand même, calculer la matrice inverse de A , ce qui est paradoxal dans la mesure où la connaissance du conditionnement de A est préalable au choix d'une précision dans le calcul (notamment, dans le but de rendre l'erreur d'arrondi négligeable devant les incertitu-

des physiques). On pourra se reporter à l'ouvrage [1] pour trouver des détails sur ce problème et la manière de le résoudre en pratique.

La signification du conditionnement est qu'il existe des matrices n'ayant pas de grands coefficients, mais dont l'inverse en possède, et qui seront détectés par une grande valeur du conditionnement. Ce sont des matrices « **presque singulières** ». L'exemple le plus connu est fourni par la matrice de Hilbert H_n ayant pour coefficient en place (i, j) le nombre $\frac{1}{i+j-1}$. Il s'agit de la matrice de la forme quadratique :

$$\begin{aligned} q(X) = X^* H_n X &= \sum_{i,j} \frac{x_i x_j}{i+j-1} = \sum_{i,j} x_i x_j \int_0^1 t^{i+j-2} dt \\ &= \int_0^1 \left(\sum_i x_i t^{i-1} dt \right)^2 \geq 0 \end{aligned}$$

et, par conséquent, sa 2-norme est égale à son rayon spectral, qui vaut aussi :

$$\mu_n = \underset{N_2(X) \leq 1}{\text{Max}} X^* H_n X$$

La 2-norme de H_n^{-1} se calcule de même et vaut l'inverse de la plus petite valeur propre de H_n , laquelle vaut encore :

$$\mu_1 = \underset{N_2(X) \leq 1}{\text{Min}} X^* H_n X$$

$$\text{et } \text{cond}_2(H_n) = \frac{\mu_n}{\mu_1}$$

On peut montrer (en se servant du théorème de la projection orthogonale) que :

$$\mu_1 \leq \frac{1}{12 \times 15^{n-2}} \text{ et que } \mu_n \geq 1,25$$

On en déduit que $\text{cond}_2(A) \geq 15^{n-1}$, ce qui est rapidement très grand. La matrice de Hilbert est donc tout indiquée comme « jeu d'essai » pour valider un algorithme d'algèbre linéaire.

1.4.2 Matrices à diagonale dominante

De nombreux problèmes d'application donnent naissance à des matrices dont les coefficients diagonaux sont assez grands ; ces matrices ont un conditionnement plutôt favorable.

Définition 10. Une matrice $A = (a_{ij})$ de taille $n \times n$ est dite à **diagonale strictement dominante** si l'on a, pour tout i :

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

Proposition 9.

Une matrice à diagonale dominante est inversible. Si l'on note :

$$\delta = \underset{1 \leq i \leq n}{\text{Min}} \left(|a_{ii}| - \sum_{j=1, j \neq i}^n |a_{ij}| \right) > 0$$

$$\text{alors } \|A^{-1}\|_\ell \leq \frac{1}{\delta}$$

Preuve. \diamond Soit un vecteur non nul $x = (x_i)$. Nous allons comparer $N_\infty(Ax)$ et $N_\infty(x)$, on a :

$$y = Ax = (y_i) \text{ avec } y_i = \sum_{j=1}^n a_{ij} x_j$$

donc :

$$|y_i| \geq |a_{ii}| |x_i| - \sum_{j \neq i} |a_{ij}| |x_j| \geq |a_{ii}| |x_i| - \left(\sum_{j \neq i} |a_{ij}| \right) N_\infty(x)$$

Choisissons la valeur de i qui maximise $|x_i|$; il vient pour cette valeur :

$$|y_i| \geq \left(|a_{ii}| - \sum_{j \neq i} |a_{ij}| \right) N_\infty(x) \geq \delta N_\infty(x)$$

Il en résulte que y n'est pas nul et que A est bien inversible.

D'autre part, on a :

$$N_\infty(y) \geq \delta N_\infty(x) = \delta N_\infty(A^{-1}y)$$

ce qui entraîne aussitôt la conclusion. \diamond

Signalons que l'étude de la dominance diagonale conduit à la localisation des valeurs propres (disques de Gershgorin).

Les matrices à dominance diagonale large sont également intéressantes, mais plus délicates à manier car des matrices comme $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, à dominance diagonale large, ne sont plus inversibles. Il faut considérer l'irréductibilité de la matrice ; on pourra se reporter à l'ouvrage [3], section 0.6, pour plus de détails.

1.5 Étude des erreurs et de leur propagation

1.5.1 Position du problème

Les problèmes étudiés en algèbre linéaire conduisent généralement à deux catégories d'algorithmes :

- les algorithmes « directs », qui consistent à multiplier successivement un vecteur initial ou une matrice initiale par des matrices connues à l'avance ;
- les algorithmes « inverses », qui consistent à résoudre en cascade une suite de systèmes linéaires, ce qui revient à multiplier par des matrices inverses.

Dans les problèmes directs, l'évaluation de la norme des matrices qui interviennent suffit généralement. Dans les problèmes inverses, le conditionnement est requis.

Considérons, par exemple, la résolution d'une équation linéaire $Ax = b$ (ce système étant supposé de Cramer, c'est-à-dire la matrice A étant inversible). Supposons, d'abord, que la matrice A soit connue avec une parfaite exactitude, mais que b soit affectée d'une erreur (due, par exemple, à une incertitude de mesure physique), de sorte qu'on n'en connaît qu'une approximation b' .

L'équation $Ax' = b'$ nous fournit une solution « perturbée » x' . Pour décider si x' est une approximation satisfaisante de la solution exacte, on peut tenter de majorer $N_2(x - x')$ ou bien considérer que x' satisfait l'équation à $N_2(b' - b)$ près de toutes façons.

Ces deux points de vue ne sont pas du tout équivalents. Par exemple, reprenons la matrice de Hilbert, en taille 4, c'est-à-dire la matrice :

$$H_4 = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix}$$

et le vecteur :

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

La solution de l'équation $H_4x = b$ vaut :

$$x = \begin{pmatrix} -4 \\ 60 \\ -180 \\ 140 \end{pmatrix}$$

Perturbons un peu b en $b' = \begin{pmatrix} 0,8 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ (on a donc $N_2(\Delta b) = 0,2$).

La solution modifiée vaut :

$$x' = \begin{pmatrix} -7,2 \\ 84 \\ -228 \\ 168 \end{pmatrix}$$

On trouve $N_2(\Delta x) = 60,615\dots$ et on voit qu'il n'y a pas de commune mesure entre la perturbation opérée sur b et celle opérée sur x . Cela dit, on peut déclarer qu'une solution approchée est satisfaisante si elle diffère peu (en norme) de la solution exacte (et ce n'est pas le cas ici), ou bien si son image diffère peu du vecteur à atteindre (et ici x' donne satisfaction de ce point de vue). La raison de ce comportement est évidemment que l'inverse de H_4 a de très grands coefficients (ils dépassent 6 000).

Il ne faut pas non plus croire que l'instabilité de H_4 se produise

dans toutes les directions. Ainsi, le vecteur $\begin{pmatrix} 1995 \\ 1134 \\ 810 \\ 630 \end{pmatrix}$, proche d'un

vecteur propre associé à la plus grande valeur propre (environ 1,5), représente-t-il une direction de stabilité.

1.5.2 Erreur sur la solution d'une équation linéaire

Considérons à présent une équation linéaire de la forme $Ax = b$, la matrice A étant supposée inversible, et dans laquelle le vecteur b est entaché d'une certaine erreur Δb et la matrice A d'une erreur ΔA ; la solution du système perturbé peut être écrite sous la forme $x + \Delta x$.

sous réserve que la matrice $A + \Delta A$ soit encore inversible. Ce dernier point peut être précisé.

Lemme 1. Soit une matrice A inversible ; pour toute matrice H telle que :

$$\|H\| < \|A^{-1}\|^{-1}$$

$A + H$ est inversible ; on a de plus :

$$(*) \quad \|(A + H)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|H\| \|A^{-1}\|}$$

et

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

Preuve. ◊ On sait que $I_n + B$ est inversible dès que $\|B\| < 1$ (par séries géométriques). Ici, si on pose $B = HA^{-1}$, on a bien :

$$\|B\| \leq \|H\| \|A^{-1}\| < 1$$

et $I + HA^{-1}$ est inversible ; donc $A + H$ l'est aussi.

Puis on constate que :

$$(A + H)^{-1} - A^{-1} = -(A + H)^{-1} HA^{-1}$$

alors :

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

ce qui implique :

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

Reportant cette majoration dans l'inégalité précédente, il vient

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

On peut reformuler cette relation en termes d'erreurs relatives :

$$\frac{\|\Delta A^{-1}\|}{\|A^{-1}\|} = \delta A^{-1} \leq \frac{\|H\| \|A^{-1}\|}{1 - \|H\| \|A^{-1}\|} = \frac{\|\Delta A\| \|A^{-1}\|}{1 - \|\Delta A\| \|A^{-1}\|} = \frac{\text{cond}(A) \delta A}{1 - \text{cond}(A) \delta A}$$

On voit clairement, dans cette formule, le rôle du conditionnement de A : s'il est grand, l'erreur relative sur l'inverse est présumée notamment plus grande que l'erreur relative sur A .

Considérons de même les erreurs relatives :

$$\delta b = \frac{\|\Delta b\|}{\|b\|} \text{ et } \delta x = \frac{\|\Delta x\|}{\|x\|}$$

Posons $H = \Delta A$. On a :

$$(A + H)(x + \Delta x) - Ax = b + \Delta b - b$$

soit encore :

$$(A + H)\Delta x = \Delta b - Hx$$

qui devient en utilisant la majoration (*) ci-dessus :

$$\begin{aligned} \|\Delta x\| &= \|(A + H)^{-1}\Delta b - (A + H)^{-1}Hx\| \leq \frac{\|A^{-1}\|}{1 - \|H\| \|A^{-1}\|} \|\Delta b\| \\ &\quad + \frac{\|A^{-1}\| \|H\|}{1 - \|H\| \|A^{-1}\|} \|x\| \end{aligned}$$

mais, de plus, on a $\|b\| \leq \|A\| \|x\|$, donc il vient :

$$\begin{aligned} \delta x &= \frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|}{1 - \|H\| \|A^{-1}\|} \frac{\|\Delta b\|}{\|x\|} + \frac{\|A^{-1}\| \|H\|}{1 - \|H\| \|A^{-1}\|} \\ &\leq \frac{\|A^{-1}\| \|A\|}{1 - \|H\| \|A^{-1}\|} \delta b + \frac{\|A^{-1}\| \|H\|}{1 - \|H\| \|A^{-1}\|} = \frac{\text{cond}(A)}{1 - \text{cond}(A)} (\delta b + \delta A) \end{aligned}$$

À partir de cette majoration, on peut prévoir la qualité de l'approximation de la solution connaissant la précision de la matrice et celle du second membre b .

1.5.3 Analyse inverse

Dans bien des situations, les algorithmes d'algèbre linéaire reviennent à multiplier successivement une matrice ou un vecteur donnés par une suite de matrices (transvections, matrices unitaires...). Dans une perspective d'application réelle, on doit considérer toutes les données numériques comme pouvant être entachées d'erreurs (de mesures, d'arrondi...).

Considérons, par exemple, le problème de la multiplication d'une matrice A par un vecteur x ; la matrice A étant erronée, on calcule en fait avec une matrice $A + \Delta A$. L'image calculée est alors :

$$(A + \Delta A)x = A(x + A^{-1}\Delta Ax) = Ax_1 ;$$

ainsi, on peut considérer que tout s'est passé comme si l'on avait appliqué la vraie matrice A à un vecteur x_1 différent de x ; la perturbation, dite **perturbation équivalente**, de ce vecteur est $A^{-1}\Delta Ax$. Si l'algorithme consiste en une suite de multiplications par des matrices inversibles, les perturbations équivalentes aux différentes erreurs vont s'ajouter et faire en sorte que tout se passe comme si l'on avait appliqué les matrices exactes sur une donnée autant de fois perturbée.

On peut appliquer la même idée pour l'analyse des erreurs d'arrondi, ainsi que pour la résolution d'une équation linéaire (qui revient à multiplier par une matrice inverse).

1.5.4 Point de vue probabiliste

Considérons à présent un vecteur aléatoire X , suivant une loi de probabilité gaussienne d'espérance M et de covariance Σ ; M est un vecteur de la même taille que X et Σ est une matrice symétrique positive, ayant pour coefficients les covariances mutuelles des composantes de X .

Rappelons que le caractère gaussien de X correspond au fait que, pour tout vecteur U , la variable aléatoire réelle U^*X suit une loi normale d'espérance U^*M et de variance $U^*\Sigma U$; la loi du vecteur X possède une densité par rapport à la mesure de Lebesgue si, et seulement si, Σ est inversible, et, dans ce cas, sa densité est :

$$f(x_1, \dots, x_n) = \frac{1}{(2\pi)^{n/2} \sqrt{\det \Sigma}} \exp\left(-\frac{1}{2} (x - M)^* \Sigma^{-1} (x - M)\right)$$

Les vecteurs aléatoires gaussiens sont une bonne modélisation des résultats de mesures physiques entachées d'incertitudes. Par exemple, si X désigne la position d'un mobile dans l'espace tridimensionnel, les incertitudes de mesure reviennent essentiellement (avec une probabilité supérieure à 0,95) à localiser X dans une ellipsoïde défini par :

$$(X - M)^* \Sigma^{-1} (X - M) \leq 3$$

Si Σ est dégénérée (non inversible), le vecteur X prend presque sûrement ses valeurs dans un plan.

Lorsqu'on applique sur X une matrice A , le vecteur $Y = AX$ résultant suit encore une loi gaussienne ; en effet, pour tout vecteur U , la variable aléatoire réelle $U^*AX = (U^*A)X$ suit une loi normale

d'espérance (U^*A) $M = U^*(AM)$ est de variance $U^*A\Sigma A^*U$; la matrice de covariance de Y est $A\Sigma A^*$.

Pareillement, si Z est solution d'une équation linéaire $AZ = X$ (A étant supposée inversible), il suivra une loi gaussienne d'espérance $A^{-1}X$ et de matrice de covariance $A^{-1}\Sigma A^{-1}$.

La « précision » de X tient essentiellement à la plus grande valeur propre (le rayon spectral) de Σ (qui correspond au plus grand axe de l'ellipsoïde d'équation $(X - M)^*\Sigma^{-1}(X - M) = \alpha^2$), qui est aussi le maximum du quotient de Rayleigh $\frac{X^*\Sigma X}{X^*X}$ (pour X non nul).

La « précision » de Y est alors associée au maximum du quotient de Rayleigh $\frac{X^*A\Sigma A^*X}{X^*X}$

Si on suppose A inversible, on peut écrire :

$$\frac{X^*A\Sigma A^*X}{X^*X} = \frac{X^*A\Sigma A^*X}{X^*AA^*X} \times \frac{X^*AA^*X}{X^*X} = \frac{Y^*\Sigma Y}{Y^*Y} \times \frac{X^*AA^*X}{X^*X}$$

avec $Y = A^*X$. Il vient donc :

$$\rho(A\Sigma A^*) \leq \rho(\Sigma) \rho(AA^*) = \rho(\Sigma) \|A\|_2^2$$

On voit alors que, dans une méthode « directe », la petitesse de A assure un contrôle de la précision du vecteur AX , tandis que, dans une méthode « inverse » (où l'on cherche $Z = A^{-1}X$), c'est la petitesse de A^{-1} qui entre en jeu.

2. Méthodes du pivot

Dans cette partie, nous décrivons une famille de méthodes qui reposent sur une technique commune, et permettent de résoudre des problèmes aussi variés que les équations linéaires, le calcul des déterminants et des matrices inverses, la recherche de relations de dépendance...

Le lecteur pourra consulter avec profit l'article [AF 85] qui présente les bases de l'algèbre linéaire.

2.1 Principe de l'échelonnement

2.1.1 But visé

Les méthodes « du pivot » forment une classe d'algorithmes d'algèbre linéaire qui agissent, en fait, sur un système de vecteurs en respectant le sous-espace engendré par celui-ci (donc aussi le rang) par le moyen de combinaisons linéaires précises. Ces méthodes reviennent à remplacer le système de vecteurs original par un système plus simple, en cela qu'il possède de nombreux coefficients nuls et organisés d'une manière commode. La phase de l'algorithme qui consiste à faire apparaître ces coefficients nuls est appelée **échelonnement**.

À la suite de l'échelonnement, diverses actions peuvent être entreprises suivant le problème étudié.

Nous considérerons donc, par la suite, une matrice :

$$A = (a_{ij})$$

ayant n lignes et p colonnes, ainsi que le système de ses vecteurs-ligne que nous noterons :

$$L_i = (a_{i1}, \dots, a_{ip}) \in \mathbb{K}^p$$

2.1.2 Opérations élémentaires

Parmi les combinaisons linéaires que l'on peut réaliser sur un système de vecteurs, il est intéressant de dégager des opérations « primitives », à partir desquelles toutes les autres peuvent se réaliser par composition.

Définition 11. Soit $\lambda \in \mathbb{K}$ un scalaire, non nul.

On appelle **dilatation** $D_i(\lambda)$ toute opération consistant à remplacer un système de vecteurs (x_1, \dots, x_n) par un système (x'_1, \dots, x'_n) tel que $x'_j = x_j$ pour tout $j \neq i$ et $x'_i = \lambda x_i$. Elle consiste à multiplier l'un des vecteur par une constante.

Soient i et k deux indices distincts et $\lambda \in \mathbb{K}$ un scalaire quelconque. On appelle **transvection** $T_{ik}(\lambda)$ toute opération consistant à remplacer (x_1, \dots, x_n) par (x'_1, \dots, x'_n) tel que :

$$x'_j = x_j \text{ pour tout } j \neq i \text{ et } x'_i = x_i + \lambda x_k$$

Elle consiste à ajouter à l'un des vecteurs un autre multiplié au passage par λ .

Les opérations élémentaires permettent de réaliser des combinaisons linéaires. En effet, remplacer un vecteur x_i par

$\sum_{1 \leq j \leq p} \lambda_j x_j$, avec $\lambda_i \neq 0$, revient à effectuer la dilatation $D_i(\lambda_i)$ suivie ou précédée de toutes les transvections $T_{ij}(\lambda_j)$ pour $j \neq i$.

Proposition 10.

Les opérations élémentaires sur les systèmes de vecteurs sont inversibles :

$$D_i(\lambda)^{-1} = D_i\left(\frac{1}{\lambda}\right)$$

et

$$T_{ij}(\lambda)^{-1} = T_{ij}(-\lambda)$$

Il se trouve que l'opération qui consiste à échanger deux vecteurs dans un système n'est pas « primitive », en cela qu'elle est réalisable par quelques opérations élémentaires. En effet, on a la proposition 11.

Proposition 11.

Étant donné deux indices i et j distincts, la composition suivante :

$$T_{ij}(1) \circ T_{ji}(-1) \circ T_{ij}(1)$$

a pour effet de transformer un système de vecteurs (x_1, \dots, x_n) en (x'_1, \dots, x'_n) tel que :

$$x'_k = x_k \text{ pour tout } k \neq i, j$$

et

$$x'_i = x_j, x'_j = -x_i$$

En conséquence, la composition :

$$D_j(-1) \circ T_{ij}(1) \circ T_{ji}(-1) \circ T_{ij}(1)$$

a pour effet d'intervertir les vecteurs x_i et x_j .

Ainsi, l'intérêt théorique des permutations sur les systèmes de vecteurs est réduit ; en pratique (en programmation), on ne permute pas non plus : c'est seulement dans un but de présentation que l'on peut souhaiter permute les vecteurs d'un système.

Ayant noté que les opérations élémentaires réalisent des combinaisons linéaires et sont réversibles, on obtient facilement la proposition 12.

Proposition 12.

Les opérations élémentaires sur les systèmes de vecteurs conservent le sous-espace engendré, donc aussi le rang.

L'aspect matriciel des opérations élémentaires peut être précisé. Représentons un système de vecteurs (L_i) dans une base (e_j) par une matrice $A = (a_{ij})$ de sorte le i -ième vecteur correspond à la i -ième ligne de la matrice. On peut aussi utiliser les colonnes mais les notations sont un peu plus simples avec les lignes, comme on va le voir. Dans ces conditions, on vérifie qu'une dilatation $D_k(\lambda)$ sur le système des lignes de A revient à remplacer A par $D_k(\lambda) A$, avec la matrice :

$$D_k(\lambda) = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \lambda & & & \\ & & & \ddots & & \\ 0 & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \quad (\text{matrice de dilatation}).$$

On peut d'ailleurs écrire :

$$D_k(\lambda) = I_n + (\lambda - 1) U_{ii}$$

en notant U_{ii} la matrice dont tous les coefficients sont nuls excepté celui qui se trouve à la ligne i et la colonne j , celui-ci valant 1.

De même, on vérifie qu'une transvection $T_{hk}(\lambda)$ sur le système des lignes de A revient à remplacer A par $T_{hk}(\lambda) A$:

$$T_{hk}(\lambda) = h \rightarrow \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \lambda & & & \\ 0 & & & \ddots & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} = I_n + \lambda U_{hk} \quad (\text{matrice de transvection}).$$

En ce qui concerne l'action sur les colonnes, on a des formules analogues, mais avec une multiplication à droite, de sorte qu'une opération consistant à ajouter à la p -ième colonne de A la q -ième multipliée par λ se réalise par $AT_{qp}(\lambda)$. On remarque ainsi l'intervention des indices par rapport à l'action sur les lignes.

2.1.3 Systèmes et matrices échelonnés

Définition 12. Soit un système $(x_i)_{1 \leq i \leq p}$ de vecteurs décrit

dans une base $(e_j)_{1 \leq j \leq q}$ par les composantes $x_i = \sum_{j=1}^q a_{ij} e_j$.

On dit que ce système est **échelonné** si le nombre de zéros consécutifs qui figurent au début de l'écriture de x_i croît strictement avec i .

On dit, de même, qu'une matrice est échelonnée si le système de ses lignes l'est.

Une matrice échelonnée a l'apparence suivante :

$$\begin{pmatrix} 0 & \cdots & 0 & m_1 & * & * & \cdots & * & \cdots & \cdots & * \\ 0 & \cdots & 0 & 0 & \cdots & 0 & m_2 & * & \cdots & \cdots & * \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & 0 & 0 & m_r & * & * & \\ 0 & \cdots & 0 & \cdots & \cdots & 0 & 0 & 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

On démontre, par un examen des composantes sur les vecteurs e_k (k étant le numéro d'un « échelon ») la proposition suivante.

Proposition 13.

Un système échelonné a pour rang le nombre de ses vecteurs lignes non nuls.

Un cas particulier apprécié de matrice échelonnée est constitué par les matrices triangulaires supérieures, ayant des coefficients diagonaux non nuls. Elles sont inversibles.

2.1.4 Méthode du pivot partiel

La méthode du pivot de Gauss s'attache par conséquent à réaliser de « bonnes » combinaisons linéaires pour aboutir à un système échelonné. Pour ce faire, elle réalise des opérations élémentaires suivant un ordre précis.

■ Première étape

On commence par chercher un « pivot », coefficient non nul du système et situé « le plus à gauche possible ».

On le trouve en colonne c_1 et en ligne ℓ_1 . Par exemple, la matrice

$$A = \begin{pmatrix} 1 & 2 & -3 & 5 & 4 \\ 2 & 4 & -6 & 3 & 1 \\ -1 & -2 & 1 & 3 & -1 \\ 0 & 0 & 1 & 3 & -5 \\ -3 & -6 & 2 & -37 & 8 \end{pmatrix}$$

conduit à prendre $p_1 = 1$, $\ell_1 = c_1 = 1$.

Le premier pivot étant localisé, on procède à l'élimination. Celle-ci consiste à combiner chaque ligne i (avec $i \neq \ell_1$) avec la ligne du pivot pour annuler le coefficient se situant dans la même colonne que le pivot. Cette combinaison est une transvection $T_{i\ell_1} \left(-\frac{a_{ic_1}}{p_1} \right)$.

La ligne du premier pivot reste inchangée.

À l'issue de cette étape, les coefficients de la colonne du premier pivot sont tous devenus nuls, excepté celui-ci, sous la forme :

$$\begin{pmatrix} 0 & * & \cdots & * \\ \vdots & & & \\ p_1 & * & \cdots & * \\ 0 & & & \\ \vdots & & & \\ 0 & * & \cdots & * \end{pmatrix}$$

$$\text{exemple : } A_2 = \begin{pmatrix} 1 & 2 & -3 & 5 & 4 \\ 0 & 0 & 0 & -7 & -7 \\ 0 & 0 & -2 & 8 & 3 \\ 0 & 0 & 1 & 3 & -5 \\ 0 & 0 & -7 & -22 & 20 \end{pmatrix} \quad \begin{aligned} L'_1 &= L_1 \\ L'_2 &= L_2 - 2L_1 \\ L'_3 &= L_3 + L_1 \\ L'_4 &= L_4 \\ L'_5 &= L_5 + 3L_1 \end{aligned}$$

Il est intéressant de relever que cette méthode, qui se veut pratique, est aussi à la base de la théorie de la dimension. Cela est expliqué dans l'article [AF 85], paragraphe 2.1.1, proposition 22.

■ Seconde étape

On cherche ensuite un second pivot, situé « le plus à gauche possible mais à droite du premier et pas dans la même ligne ».

On le trouve (éventuellement) en colonne c_2 et en ligne ℓ_2 , avec $c_2 > c_1$ et c_2 minimale, $\ell_2 \neq \ell_1$. Le fait d'avoir $c_2 > c_1 + 1$ signifie que tous les coefficients sont nuls dans les colonnes situées entre c_1 et c_2 et les lignes autres que ℓ_1 .

Ne pas trouver le second pivot signifie que le travail est terminé ; on passe alors à la fin de l'algorithme.

Le second pivot étant localisé, on procède à l'élimination exactement comme à la première étape, à ceci près qu'on laisse la ligne du premier pivot toujours inchangée. La matrice a à ce moment l'allure suivante :

$$\begin{pmatrix} 0 & 0 & * & \cdots & * \\ \vdots & \vdots & & & \\ p_1 & 0 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & p_2 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \end{pmatrix}$$

exemple :

$$\ell_2 = 4, c_2 = 3, A_2 = \begin{pmatrix} 1 & 2 & -3 & 5 & 4 \\ 0 & 0 & 0 & -7 & -7 \\ 0 & 0 & 0 & 14 & -7 \\ 0 & 0 & 1 & 3 & -5 \\ 0 & 0 & 0 & -1 & -15 \end{pmatrix} \begin{array}{l} L'_1 = L'_1 \\ L'_2 = L'_2 \\ L''_3 = L'_3 + 2L'_4 \\ L''_4 = L'_4 \\ L''_5 = L'_5 + 7L'_4 \end{array}$$

Poursuite

On traite ainsi toutes les colonnes de gauche à droite (c'est pourquoi on dit que c'est une stratégie de pivotage partiel) tant que le choix d'un nouveau pivot reste possible, dans une ligne non encore choisie.

Fin de l'algorithme

À l'issue de l'algorithme, on a épuisé toutes les colonnes de la matrice dans la recherche des pivots. Il se peut que le système résultant contienne des lignes nulles ; cela signifie simplement que le système initial avait des lignes linéairement dépendantes. En fin de compte, on a trouvé une matrice échelonnée à l'ordre des lignes près. Elle peut devenir échelonnée si on permute ses lignes de manière à placer la ligne ℓ_1 en première ligne, la ligne ℓ_2 en seconde, et ainsi de suite. Dans ces conditions, le rang n'a pas changé tout au long de l'algorithme et la matrice initiale a pour rang le nombre de pivots.

L'exemple proposé se termine avec :

$$A = \begin{pmatrix} 1 & 2 & -3 & 5 & 4 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & -5 \\ 0 & 0 & 0 & -1 & -5 \end{pmatrix}$$

(de rang 4 ; les pivots sont en gras).

2.1.5 Bilan matriciel de la méthode

Chaque transvection réalisée sur A revient à la multiplier à gauche par une matrice inversible. Supposons que la permutation des lignes soit faite sur A avant de commencer l'algorithme (et non à la fin) ; dans ces conditions, le choix des pivots se fera immédiatement (le k -ième pivot en ligne k) et une matrice échelonnée sera produite. Dans cette optique, les combinaisons linéaires effectuées sur les lignes consistent toujours à combiner une ligne avec une ligne de pivot située au-dessus d'elle, et la matrice de transvection

$T_{i\ell_1} \left(-\frac{a_{i\ell_1}}{p_1} \right)$ devient triangulaire inférieure.

On vient donc de démontrer la décomposition suivante :

$$MQA = R,$$

M étant la matrice triangulaire inférieure qui résulte du produit des diverses matrices de transvections et Q une matrice de **permutation** (ayant un seul coefficient non nul, valant 1, dans chaque ligne et chaque colonne).

Prenant les inverses de M et de Q on trouve le résultat théorique suivant.

Théorème 2. Pour toute matrice $A \in \mathbb{M}_{pq}(\mathbb{K})$, il existe une matrice de permutation P , une matrice triangulaire inférieure L , ayant des 1 sur sa diagonale, et une matrice échelonnée R de même taille que A , telles que :

$$A = PLR$$

En particulier, si A est une matrice carrée, alors R est une matrice triangulaire supérieure.

Corollaire 6.

Toute matrice A carrée inversible peut être décomposée en :

$$A = PLU$$

P et L étant conformes à la proposition précédente, et U étant une matrice triangulaire supérieure inversible.

2.1.6 Stratégies de pivotage et complexité

Numériquement parlant, prendre pour pivot le premier coefficient non nul serait certainement dangereux, car la division par ce nombre risque de faire apparaître au mieux de grandes erreurs d'arrondi, au pire des débordements de capacité. On limite quelque peu ce risque en choisissant, dans chaque colonne, pour pivot le coefficient de plus grand module (toujours situé hors des lignes des pivots précédents).

Une amélioration ultérieure de cette idée consiste à effectuer un « pivotage total », c'est-à-dire à ne plus chercher les pivots de gauche à droite mais à choisir, à chaque étape, comme pivot le coefficient de plus grand module de toute la matrice, situé hors des lignes des pivots précédents. Cela complique un peu l'algorithme et le ralentit (par la nécessité de nombreuses comparaisons), tout en diminuant les risques de dépassement de capacité.

De toutes manières, on peut démontrer que la complexité algorithmique (le nombre d'opérations arithmétiques requises pour exécuter l'algorithme dans le pire des cas) de la méthode du pivot menée sur une matrice A carrée de taille n est de l'ordre de grandeur de $a \cdot n^3$, a étant une constante, lorsque n est assez grand.

Ce résultat est difficilement améliorable avec des moyens simples, sachant que le produit de deux matrices carrées de taille n « consomme » aussi environ n^3 calculs.

2.1.7 Algorithme

Nous pouvons à présent schématiser l'algorithme d'échelonnement.

Variables :

une matrice $A = (a_{ij})$ ayant n lignes et p colonnes
piv contenant le pivot

h, i, j, k, r des indices

lp un vecteur (tableau) ayant p composantes (lignes de pivot).

Initialisations :

au début, $k = 0$, $r = 0$, $piv = 0$ et lp rempli de zéros.

Itération :

tant que $k \leq p$ faire :

- augmenter k de 1
- chercher un pivot $piv = a_{hk}$ avec $lp_h = 0$ et $|piv|$ maximal
- si le pivot est trop petit, stopper (problème mal conditionné)
- si le pivot est trouvé, lancer l'élimination
- et continuer

et passer à la fin de l'algorithme.

Procédure d'élimination pour un pivot donné en ligne h , colonne k :

augmenter r de 1

poser $lp_h = r$

faire varier i de 1 à n :

si $lp_i > 0$ éliminer la ligne i

et retourner.

Procédure d'élimination d'une ligne de numéro i pour un pivot donné en ligne h , colonne k :

faire varier j de 1 à k :

remplacer a_{ij} par 0

faire varier j de $k+1$ à n :

remplacer a_{ij} par $a_{ij} - \frac{a_{ik}a_{hj}}{piv}$

et retourner.

Fin de l'algorithme :

A contient la matrice échelonnée à l'ordre des lignes près
 r contient le nombre de pivots (rang)

lp contient les numéros des lignes des pivots

piv contient le dernier pivot.

Pour obtenir une matrice B véritablement échelonnée, il suffit de l'initialiser à 0 et recopier la ligne lp_i de A dans la ligne i de B , ceci pour tout $i \leq r$. Bien entendu, de manière générique, le rang sera égal à $\text{Min}(n, p)$. Autrement, on a des relations de dépendance linéaires qui apparaissent et elles n'ont de signification qu'en l'absence d'erreurs d'arrondi et de mesure, dans un contexte de calcul exact et non approché.

2.2 Une variante : la méthode de Crout (dite LU)

2.2.1 Principe de la méthode

La plupart des matrices qu'on rencontre en pratique sont carrées, inversibles, et l'application de la méthode du pivot partiel se fait sans aucune nécessité d'échange de lignes. Cela provient de deux phénomènes.

D'une part, on a souvent affaire à des matrices à diagonale strictement dominante.

Proposition 14.

Si A est une matrice carrée à diagonale strictement dominante, la méthode du pivot peut être menée sur A en choisissant systématiquement les pivots sur la diagonale ; la matrice triangulaire R obtenue à la fin de l'algorithme d'échelonnement est encore à diagonale strictement dominante.

Preuve. ◇ On procède par récurrence : le premier pivot peut évidemment être pris en première ligne et en première colonne ; il suffit de vérifier que la première étape de la méthode du pivot respecte

la dominance diagonale stricte. Or, si c'est le cas pour A et que l'on calcule :

$$a'_{ij} = a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}} \text{ avec } a'_{i1} = 0$$

alors :

$$\begin{aligned} \sum_{j \neq i} |a'_{ij}| &= \sum_{j \neq i, j > 1} |a'_{ij}| \leqslant \sum_{j \neq i, j > 1} |a_{ij}| + \frac{|a_{i1}|}{|a_{11}|} \sum_{j \neq i, j > 1} |a_{1j}| \\ &< |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} (|a_{11}| - |a_{1i}|) \\ &= |a_{ii}| - \frac{|a_{i1}||a_{1i}|}{|a_{11}|} \leqslant |a_{ii}| \end{aligned} \diamond$$

■ D'autre part, l'obligation de chercher un pivot non nul ailleurs qu'au premier endroit possible correspond à la nullité de certains sous-déterminants (mineurs) de la matrice initiale, situation exceptionnelle et numériquement improbable (sauf avec des coefficients entiers et se situant dans un ensemble assez petit). Dans ces conditions, une telle matrice va se factoriser sous la forme :

$$A = LU$$

Plus précisément, on a le théorème 3.

Théorème 3. Une matrice $A = (a_{ij})$ carrée inversible peut se factoriser en :

$$A = LU,$$

L étant une matrice triangulaire inférieure ayant des 1 sur sa diagonale, et U une matrice triangulaire supérieure inversible, si, et seulement si, toutes les sous-matrices carrées $A_k = (a_{ij})_{1 \leq i, j \leq k}$ sont inversibles. Lorsque c'est le cas, la décomposition est unique.

Preuve. ◇ On procède par récurrence, écrivant les matrices par blocs sous la forme

$$A = \begin{pmatrix} A' & V \\ W & \alpha \end{pmatrix}, \quad L = \begin{pmatrix} L' & 0 \\ H & 1 \end{pmatrix}, \quad U = \begin{pmatrix} U' & S \\ 0 & \gamma \end{pmatrix} \diamond$$

2.2.2 Pratique

Les coefficients des matrices L et U peuvent être déterminés de proche en proche par des équations linéaires.

On connaît d'abord $1 \cdot u_{1j} = a_{1j}$, puis $\ell_{k1}u_{11} = a_{k1}$ donne tous les ℓ_{kj} ; ce qui permet de passer à la seconde colonne par $\ell_{21}u_{12} + 1 \cdot u_{2j} = a_{2j}$; comme ℓ_{21} est déjà connu, les coefficients u_{2j} se déterminent et permettent à la suite la détermination des ℓ_{ij} selon $\ell_{11}u_{12} + \ell_{12}u_{22} = a_{12}$. On voit, à l'occasion, qu'il est nécessaire de diviser par les coefficients diagonaux de U qui jouent exactement le rôle des pivots ; les risques d'instabilité inhérents à la méthode du pivot sont donc présents. Ces coefficients sont tels que :

$$\det A_k = \prod_{i=1}^k u_{ii} \text{ (ceci se voit par un produit par blocs),}$$

soit :

$$u_{kk} = \frac{\det A_k}{\det A_{k-1}}$$

Il est clair que l'instabilité se produira si le déterminant de A est petit, mais elle peut aussi survenir avec un déterminant $\det A_k$ petit compensé par $\det A_{k+1}$ grand.

2.3 Application à la résolution des systèmes linéaires

Considérons une équation linéaire de la forme $AX = B$, X et B étant des matrices pourvues d'une seule colonne (A n'étant pas nécessairement carrée). Par la méthode du pivot, nous pouvons obtenir :

$$SMA = R$$

S étant une matrice de permutation, M une matrice inversible (produit de matrices de transvections) et R une matrice échelonnée.

L'équation proposée revient à :

$$RX = SMB$$

Pour calculer SMB , on applique sur B les mêmes opérations élémentaires et permutations de lignes que sur A .

Rappelons qu'en pratique il n'est pas nécessaire de permuter les lignes, il suffit de connaître les numéros des lignes contenant les pivots. Tout revient donc à travailler sur une matrice A' ayant une colonne de plus, obtenue en y recopiant le vecteur B :

$$A' = (A \mid B) \text{ entraîne } SMA' = (SMA \mid SMB) = (R \mid C).$$

Le système ainsi obtenu, à savoir $RX = C$, est « échelonné » et peut se résoudre de proche en proche sans difficulté (en commençant par l'équation correspondant à la ligne du dernier pivot).

Si la matrice A n'est pas inversible, il peut y avoir strictement moins de pivots que de lignes ; dans ce cas, la matrice R a des lignes entièrement nulles. Si les coefficients correspondants du vecteur C ne sont pas nuls, le système n'aura aucune solution (il est alors dit incompatible). Sinon, on peut ignorer ces lignes. S'il y a moins de pivots que de colonnes, les inconnues ne peuvent pas être toutes déterminées ; on peut assigner aux « inconnues » (composantes de X) dont les numéros ne sont pas des numéros de colonne des pivots (on les appelle des **inconnues non principales**) des valeurs arbitraires. Cela fait, le système restant, ayant pour inconnues les composantes de X qui correspondent aux numéros de colonne des pivots (on les appelle des **inconnues principales**), est essentiellement un système triangulaire et se résout de proche en proche.

De manière équivalente, la méthode de Crout conduit à une factorisation $A = LU$; la résolution de $AX = B$ revient à la résolution successive des systèmes triangulaires :

$$LY = B \text{ puis } UX = Y$$

2.4 Calcul de déterminants

La méthode du pivot permet le calcul des déterminants des matrices carrées. Notons d'abord que la méthode de choix des pivots repose sur le rang de la matrice ; si ce rang n'est pas maximal, la matrice est singulière et son déterminant est nul. Sinon, on doit pouvoir trouver un pivot dans chaque colonne. En pratique, un déterminant très petit peut se confondre avec des erreurs d'arrondi (voir la matrice de Hilbert étudiée au paragraphe au 1.5.1) ; dans un tel cas, il convient de s'abstenir (une matrice presque singulière peut provenir d'une matrice singulière qui a subi des erreurs d'arrondi ou de mesure).

Les transvections ne modifient pas le déterminant, il apparaît que, dans la méthode d'échelonnement qui conduit à $MQA = R$, on a :

$$\det(MQA) = \det Q \det A = \det R$$

Comme R est échelonnée et inversible, son déterminant est le produit de ses coefficients diagonaux, donc égal au produit des pivots. Notons σ la permutation qui permet de remettre la ligne du premier pivot en première ligne, et ainsi de suite ; on a donc :

$$\sigma = \begin{pmatrix} i_1 & i_2 & \dots & i_n \\ 1 & 2 & \dots & n \end{pmatrix} \text{ et } \sigma(i_k) = k$$

Dès lors, le déterminant de Q coïncide avec la signature de σ (notée ici $\varepsilon(\sigma)$), soit 1 si le nombre d'échanges requis pour réaliser σ est pair et -1 sinon.

On a finalement à l'issue de la méthode du pivot :

$$\det A = \varepsilon(\sigma) \prod_{i=1}^n p_i$$

La signature de la permutation σ se calcule facilement, au cours de la recherche des pivots. En effet, on a la formule :

$$\varepsilon(\sigma) = \prod_{k>j} \operatorname{sgn} (\sigma(k) - \sigma(j))$$

Chaque fois que l'on choisit un nouveau pivot (et sa ligne $\sigma(k)$), il suffit de connaître $\prod_{j=1}^{k-1} \operatorname{sgn} (\sigma(k) - \sigma(j))$, ce qui résulte d'un simple examen, terme à terme.

Le résultat précédent montre aussi quels sont les coefficients que la méthode du pivot calcule. En effet, si nous considérons pour $i \notin \{\ell_1, \dots, \ell_k\}$ et $j \notin \{c_1, \dots, c_k\}$ la matrice :

$$B_k = \begin{pmatrix} a_{\ell_1 c_1} & a_{\ell_1 c_2} & \dots & a_{\ell_1 c_k} & a_{\ell_1 j} \\ a_{\ell_2 c_1} & a_{\ell_2 c_2} & \dots & a_{\ell_2 c_k} & a_{\ell_2 j} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ a_{\ell_k c_1} & a_{\ell_k c_2} & \dots & a_{\ell_k c_k} & a_{\ell_k j} \\ a_{i c_1} & a_{i c_2} & \dots & a_{i c_k} & a_{i j} \end{pmatrix} \in \mathcal{M}_{k+1}(\mathbb{K})$$

on peut lui appliquer l'algorithme de Gauss en choisissant les pivots systématiquement sur la diagonale, avec exactement les mêmes calculs arithmétiques (puisque ce sont, en fait, les mêmes coefficients, en des places différentes). À la fin, on aura :

$$\det A_k = p_1 p_2 \dots p_k$$

et

$$\det B_k = p_1 p_2 \dots p_k a_{ij}^{(k)}$$

$a_{ij}^{(k)}$ désignant le coefficient de A calculé à la place (i, j) à l'issue de l'élimination par le k -ième pivot p_k . Finalement, on a :

$$a_{ij}^{(k)} = \frac{\det B_k}{\det A_k}$$

et B_k comme A_k sont, à l'ordre des lignes près, des sous-matrices de la matrice initiale.

Ce résultat montre que l'instabilité numérique de la méthode du pivot est essentiellement due au fait que les mineurs de taille k peuvent être tous simultanément petits, sans que le déterminant de A lui-même soit petit.

2.5 Obtention de la matrice inverse par la méthode de Gauss-Jordan

Soit une matrice A carrée inversible, dont il s'agit de calculer l'inverse. On peut modifier légèrement la méthode du pivot pour faciliter la recherche de matrices inverses. Pour ce faire, on poursuit l'élimination à chaque étape sur les lignes des pivots antérieurs. Au lieu d'aboutir à une matrice triangulaire supérieure U , il en résulte une matrice diagonale (avec les pivots sur la diagonale), qui est très facilement inversible. En somme, si on calcule

$$SMA = U$$

(S étant la matrice de permutation qui remet les lignes des pivots en bon ordre et rend U exactement diagonale), et si U est diagonale, alors :

$$A = M^{-1} S^{-1} U$$

et

$$A^{-1} = U^{-1} SM$$

Il reste à savoir calculer M . Pour ce faire, on « accolle » à la matrice A une matrice identité I_n de même taille pour en faire une matrice $A' = (A | I_n)$, de taille $n \times 2n$. On travaille alors sur cette matrice. Il vient :

$$SMA' = (SMA | SM) = (U | SM).$$

Il suffit donc de multiplier les lignes de la partie droite SM par les inverses des pivots pour trouver la matrice inverse de A .

Sur le plan informatique, la structure de A' est assez dispendieuse et on peut économiser beaucoup de place. Voir [4] à ce sujet.

2.6 Recherche de relations de dépendance

La méthode du pivot peut être adaptée pour déterminer des relations de dépendance linéaire pouvant avoir lieu sur un système de vecteurs liés. Ce travail n'est pas fondièrement différent de ceux qui ont été décrits ci-dessus, mais n'a de sens que dans la mesure où tous les calculs peuvent être menés de manière absolument exacte (sans aucune erreur d'arrondi). On pourra se reporter à l'ouvrage [11], page 111, pour plus de détails.

2.7 Faut-il faire confiance à la méthode du pivot ?

En pratique, les calculs arithmétiques avec les nombres représentables dans un ordinateur ne sont pas identiques à ceux qui devraient être. Ou bien on tronque les représentations (principe de la « virgule flottante ») et les calculs sont faux, ou bien on dispose d'une précision illimitée et alors la mémoire étant limitée certains calculs ne pourront pas aboutir.

Le problème avec le calcul du rang est que toute erreur d'arrondi est unacceptable, car elle détruit les combinaisons linéaires ; comme le rang est un nombre entier, on peut dire que le résultat est potentiellement faux en cas d'erreurs d'arrondi. La seule solution pour déterminer le rang d'un système de vecteurs est de recourir à un logiciel de calcul symbolique capable de précision « illimitée ».

Signalons qu'il existe une variante de la méthode du pivot (Gauss et Gauss-Jordan), dite **méthode de Bareiss**, qui au prix d'un nombre plus élevé de multiplications permet de respecter complètement le caractère entier des coefficients de la matrice initiale (si cela se pré-

sente). Une telle méthode est adéquate pour rechercher des rangs et des relations de dépendance sans risque d'erreur (il suffit de tester les dépassements de capacité) ; voir [11] page 107, [12] page 48, [18] page 63, et [19].

Pour ce qui concerne les calculs de matrices inverses et des solutions d'équations linéaires, la méthode du pivot de Gauss a une certaine propension à amplifier les erreurs d'arrondi, ce qui lui donne une réputation d'instabilité numérique. Bien entendu, cet inconvénient ne se présente que pour des systèmes de grande taille ou mal conditionnés. C'est pourquoi on préfère, en général, des méthodes euclidiennes (exposées au paragraphe 4.2 : QR, Cholesky, etc.) qui sont moins à la merci des erreurs d'arrondi et de la croissance exacerbée des coefficients.

Retenons pourtant que la méthode du pivot reste la plus simple et aisément réalisable des méthodes de l'algèbre linéaire pour des **problèmes de petite taille** et suffisamment bien conditionnés ; son intérêt est, d'autre part, marqué pour des problèmes de calcul exact (ou formel) dans lesquels la rationalité des calculs est importante, alors que l'usage de fonctions transcendentales comme racine carrée, cosinus ou tangente, n'est pas bienvenu.

On peut appliquer la méthode du pivot dans des cadres particulièrement favorables, avec les matrices tridiagonales ou encore de Hessenberg.

3. Méthodes itératives

Dans ce chapitre on considère à nouveau le problème de la résolution d'une équation linéaire de la forme $Ax = b$, A étant une matrice inversible, mais au lieu de chercher un algorithme permettant d'obtenir la solution directement, si elle existe, on examine des procédés d'itérations qui peuvent créer une suite convergente de vecteurs (x_n) , ayant pour limite la solution désirée.

Ce type de méthode est très utilisé pour des **problèmes de grande taille**, lorsque la matrice A contient beaucoup de coefficients nuls hors de sa diagonale.

On pourra se reporter au chapitre 10 de [9] et au chapitre 7 de [1] pour plus de détails.

3.1 Principes théoriques

3.1.1 Méthodes des approximations successives

Nous pouvons remplacer l'équation $Ax = b$ par une équation équivalente :

$$PAx = Pb$$

P étant une matrice inversible qui sera choisie ultérieurement, puis par un problème de point fixe de la forme :

$$x = x + (Pb - PAx) = (I_n - PA)x + Pb = F(x)$$

La méthode des approximations successives (expliquée dans l'article A 100, paragraphe 6.4) suggère alors de s'intéresser à l'itération de la fonction F ; plus précisément, si on note F^k la k -ième composée successive de F avec elle-même, on considère la suite de vecteurs :

$$x_k = F(x_{k-1}) = F^k(x_0)$$

le vecteur x_0 restant à choisir.

Si cette suite converge vers un vecteur y , la continuité de F assure que l'on a $F(y) = y$, c'est-à-dire une solution au problème étudié.

Deux problèmes se posent alors :

- choisir une fonction F (donc une matrice P) assurant une convergence du processus ;
- estimer la qualité de l'approximation y par x_k en fonction de A , b et k .

3.1.2 Étude d'une itération affine

Considérons donc le problème de l'itération d'une fonction affine vectorielle :

$$F(x) = Mx + c$$

M étant une matrice carrée de taille n et c un vecteur (fixe) à n composantes.

Définition 13. L'itération définie par $x_{k+1} = F(x_k)$ est dite **consistante** par rapport à l'équation linéaire $Ax = b$ si elle n'a pas d'autre point fixe de $A^{-1}b$; elle est dite **convergente** si elle converge quel que soit le vecteur x_0 .

Proposition 15.

L'itération définie par $x_{k+1} = Mx_k + c$ est consistante par rapport à l'équation linéaire $Ax = b$ si, et seulement si, la matrice $I - M$ est inversible et que :

$$c = (I - M) A^{-1}b$$

Dans ce cas, elle est convergente si, et seulement si, le rayon spectral de M vérifie $\rho(M) < 1$ (ce qui oblige $I - M$ à être inversible).

Preuve. \diamond Si $I - M$ n'est pas inversible, elle a un noyau non réduit à 0, donc la fonction F possède plusieurs points fixes, rendant l'itération non consistante. Sinon, F a un unique point fixe. La solution $A^{-1}b$ de l'équation $Ax = b$ est ce point fixe si, et seulement si :

$$F(A^{-1}b) = MA^{-1}b + c = A^{-1}b$$

ce qui revient à :

$$c = (I - M) A^{-1}b$$

Supposons l'itération consistante. Posons $v = A^{-1}b$. On a :

$$F(x) - F(v) = Mx + c - Mv - c = M(x - v)$$

par conséquent, on a :

$$x_k - v = M^k(x_0 - v)$$

Cette suite converge, pour tout vecteur x_0 , vers 0 si, et seulement si, la suite de matrices M^k tend vers 0. On a vu au paragraphe 1.3.1 (corollaire 1) que cette condition équivaut à ce que le rayon spectral de A soit strictement inférieur à 1. \diamond

Il est particulièrement utile de noter que la convergence s'établit, dans ce cas, indépendamment du vecteur initial x_0 . Cela dispense quasiment de mener une discussion sur l'accumulation des erreurs d'arrondi, si la matrice A n'est pas affectée de telles erreurs. En effet, comme on l'a fait voir au paragraphe 1.5.3, l'effet des erreurs d'arrondi dans le calcul de x_k à partir de x_0 revient à remplacer le vecteur x_0 par un autre x'_0 (selon la « perturbation équivalente ») ; comme la limite de l'itération ne dépend pas du vecteur initial, remplacer x_0 par x'_0 n'altère pas, en principe, la qualité du résultat.

Pratiquement parlant, il convient de savoir majorer l'erreur commise en approchant v par x_k ; le plus simple est d'écrire :

$$N_\infty(x_k - v) \leq \|M^k\|_\ell N_\infty(x_0 - v)$$

v n'est pas à priori connu, mais on peut estimer sa norme grâce au conditionnement de A . On comprend dès lors que l'estimation de

l'erreur dépend de l'estimation de $\|M^k\|_\ell$. Or, on a vu au paragraphe 1.3.1 que $\|M^k\|_\ell \geq (\rho(A))^k$ et qu'il existe une constante μ et un entier q tels que $\|M^k\|_\ell \sim \mu k^q (\rho(A))^k$ lorsque k tend vers l'infini. Bien que le calcul exact de $\|M^k\|_\ell$ ne soit pas avantageux, on peut en estimer la grandeur et ce résultat sera d'autant plus favorable que le rayon spectral sera petit. Il reste à déterminer des choix de P permettant de rendre le rayon spectral de $M = I - PA$ aussi petit que possible.

Notation. Pour toute la fin de ce paragraphe, étant donnée une matrice $A = (a_{ij})$ carrée, nous noterons :

- D la matrice diagonale ayant mêmes coefficients diagonaux que A ;
- $E = (e_{ij})$ la matrice strictement triangulaire inférieure ayant mêmes coefficients que $-A$ au-dessous de la diagonale :

$$e_{ij} = \begin{cases} 0 & \text{si } i \leq j \\ -a_{ij} & \text{si } i > j \end{cases}$$

• F la matrice strictement triangulaire supérieure ayant mêmes coefficients que $-A$ au-dessus de la diagonale :

$$f_{ij} = \begin{cases} 0 & \text{si } i \geq j \\ -a_{ij} & \text{si } i < j \end{cases}$$

On a, dans ces conditions :

$$A = D - E - F$$

Nous supposerons désormais que les coefficients diagonaux de A sont non nuls, donc que D est inversible.

Cette hypothèse est notamment vérifiée si A est à diagonale strictement dominante.

3.2 Méthode de Gauss-Jacobi

Choisissons :

$$P = D^{-1} \quad (\text{aisément calculable})$$

et

$$M = I_n - D^{-1}A = D^{-1}(E + F)$$

Le vecteur c se calcule aisément :

$$c = D^{-1}b = \left(\frac{b_i}{a_{ii}} \right)$$

l'itération est alors consistante.

L'algorithme de Gauss-Jacobi consiste à calculer à partir d'un vecteur quelconque x_0 la suite :

$$x_{k+1} = Mx_k + c$$

c'est-à-dire, si $x_k = (x_i^{(k)})$ alors :

$$x_i^{(k+1)} = - \sum_{j \neq i} \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}$$

Proposition 16.

Si A est une matrice à diagonale strictement dominante, la méthode de Gauss-Jacobi pour le calcul de la solution de $Ax = b$ est convergente.

Preuve. ◊ La matrice $M = D^{-1}(E + F)$ se calcule aisément ; elle a pour coefficients :

$$m_{ii} = 0 \text{ et } m_{ij} = -\frac{a_{ij}}{a_{ii}} \text{ pour } i \neq j$$

On a donc :

$$\sum_{j=1}^n |m_{ij}| = \left| \sum_{j \neq i} \frac{a_{ij}}{a_{ii}} \right| \leq \frac{\sum_{j \neq i} |a_{ij}|}{|a_{ii}|} < 1$$

par hypothèse de dominance diagonale ; il en résulte que :

$$\|M\|_\ell < 1$$

et donc que :

$$\rho(A) < 1$$

ce qui entraîne le résultat. ◊

De plus, la convergence se produit au moins à la vitesse de $\|M^k\|_\ell \leq \|M\|_\ell^k$, et il est facile d'estimer cette dernière norme. Si la dominance diagonale n'est pas assurée, on peut tenter d'équilibrer la matrice pour atteindre une telle dominance, en la multipliant à gauche et à droite par des matrices diagonales bien choisies.

En pratique, le calcul d'une itération de la méthode de Gauss-Jacobi a un niveau de complexité (nombre de calculs arithmétiques) de l'ordre de grandeur de n^2 si tous les produits doivent être calculés ; ce n'est donc une méthode intéressante que si un grand nombre de coefficients de A sont nuls (matrice creuse) ou si la convergence est suffisamment rapide pour nécessiter nettement moins de n itérations (puisque la méthode directe du pivot donne une réponse avec un nombre de calculs de l'ordre de grandeur de n^3).

3.3 Méthode de Gauss-Seidel

La méthode de Gauss-Jacobi a un inconvénient : pour procéder au calcul de x_{k+1} il faut disposer de x_k tout au long des calculs, ce qui nécessite un stockage temporaire (à moins de recourir à un calculateur parallèle). Comme il s'agit de méthodes qui ont pour vocation de s'attaquer à des problèmes de grande taille, il est souhaitable de trouver une méthode moins encombrante. L'idée est d'utiliser au fur et à mesure les composantes de x_{k+1} qui viennent d'être calculées. Ce qui conduit aux formules :

$$x_i^{(k+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}$$

Un tel schéma de calcul n'utilise plus de stockage temporaire des coefficients. La mise en forme matricielle de cet algorithme est par contre un peu plus subtile. Nous pouvons réécrire la formule précédente comme :

$$x_i^{(k+1)} + \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(k+1)} = -\sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} + \frac{b_i}{a_{ii}}$$

ce qui s'écrit aussi :

$$(I - D^{-1} E) x_{k+1} = D^{-1} F x_k + D^{-1} b$$

ce qui revient à avoir :

$$\begin{aligned} x_{k+1} &= (I - D^{-1} E)^{-1} D^{-1} F x_k + (I - D^{-1} E)^{-1} D^{-1} b \\ &= (D - E)^{-1} F x_k + (D - E)^{-1} b \end{aligned}$$

Mais il n'est pas nécessaire de calculer l'inverse de $D - E$, comme les formules d'itération ci-dessus le prouvent. On démontre, presque comme pour la méthode de Gauss-Jacobi, la proposition suivante.

Proposition 17.

Si A est une matrice à diagonale strictement dominante, la méthode de Gauss-Seidel pour le calcul de la solution de $Ax = b$ est convergente.

On peut aussi montrer que ce processus est convergent lorsque la matrice A est symétrique réelle (ou hermitienne complexe) définie positive.

3.4 Généralisation

Il existe une méthode, dite de **relaxation**, qui généralise la méthode de Gauss-Seidel. Elle consiste à choisir un coefficient α positif et à calculer à la k -ième étape :

$$\begin{aligned} x_i^{(k+1)} &= \frac{\alpha}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) + (1 - \alpha) x_i^{(k)} \\ &= x_i^{(k)} + \frac{\alpha}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) \end{aligned}$$

(c'est une moyenne pondérée). Cette méthode est décrite dans les ouvrages [1] et [3], où le choix de α est longuement discuté (généralement voisin de 1).

On trouvera au paragraphe 4 relatif aux méthodes euclidiennes, l'importante méthode du gradient (ou de la plus profonde descente) pour la résolution d'une équation linéaire.

3.5 Amélioration itérative des solutions

Le concept de l'itération trouve aussi une application en aval des méthodes de résolution d'équations linéaires. Supposons que nous disposions d'une équation linéaire $Ax = b$ à résoudre (A étant supposée inversible) et d'une méthode numérique fournissant une solution x_1 (approchée) pour cette équation.

On peut constater que x_1 n'est pas une solution exacte en calculant :

$$\varepsilon_1 = -Ax_1 + b$$

ce vecteur est normalement très petit ; il importe donc, dans son calcul, de connaître A , b et x_1 avec une précision suffisante.

Pour corriger x_1 , on peut considérer l'équation linéaire $Ad = \varepsilon_1$; elle aura une solution $d = A^{-1}\varepsilon_1$ que notre méthode numérique remplacera par d_1 .

En conséquence, $x_2 = x_1 + d_1$ peut passer pour une meilleure approximation de la solution cherchée.

On peut recommencer le processus autant de fois que l'on veut. Cependant, l'étude de la convergence de ce processus itératif nécessite une analyse fine de la propagation des erreurs d'arrondi, que l'on trouvera dans l'ouvrage [2], chapitre V, paragraphe 38.

4. Méthodes euclidiennes

Ce paragraphe introduit des méthodes de nature géométrique pour résoudre les problèmes d'équations linéaires, tant d'un point

de vue exact qu'approché (à travers le concept de pseudo-solutions). Les méthodes de Jacobi, de la bisection et du QR itéré sont destinées au calcul des valeurs propres et figurent dans un autre article.

Nous considérons un espace vectoriel E euclidien réel ou complexe de dimension n , c'est-à-dire, un espace vectoriel muni d'un produit scalaire et de la norme qui s'en déduit. Nous noterons $(x|y)$ le produit scalaire de deux vecteurs x et y et $\|x\|$ la norme d'un vecteur x (c'est en fait $N_2(x)$). Si une base orthonormale (e_1, \dots, e_n) de E est disponible, le produit scalaire se calcule comme

$$(x|y) = \sum_{i=1}^n \bar{x}_i y_i, \text{ et matriciellement cela s'écrit } (x|y) = X^* Y, \quad X^*$$

désignant la matrice conjuguée de la transposée de X (elle a une seule ligne ici).

4.1 Un peu de théorie

4.1.1 Principe des projections orthogonales

Soit un sous-espace vectoriel F de E . Définissons le sous-espace **orthogonal** (ou polaire) F^\perp de F dans E comme l'ensemble des vecteurs orthogonaux à tous les vecteurs de F .

Rappelons que la distance $d(x, F)$ d'un élément x de E à F est, par définition, la borne inférieure des distances $d(x, z)$ lorsque z décrit F .

Théorème de la projection orthogonale. Soit un sous-espace vectoriel F de dimension finie p de E . Alors :

- (i) F admet au moins une base orthonormale (e_1, e_2, \dots, e_p) ;
- (ii) F^\perp est un supplémentaire de F , aussi appelé supplémentaire orthogonal de F ;
- (iii) le projecteur sur F parallèlement à F^\perp est noté p_F ; on appelle **projection orthogonale** de x sur F le vecteur $p_F(x)$; il se calcule grâce à une base orthonormale (e_1, e_2, \dots, e_p) de F par la formule :

$$p_F(x) = \sum_{j=i}^p (e_j|x)e_j$$

- (iv) on a $\dim F^\perp = \dim E - \dim F$ et $F^{\perp\perp} = F$;
- (v) la fonction qui, à tout élément z de F associe $\|x-z\| = d(x, z)$ atteint son minimum en un point et un seul, à savoir $p_F(x)$; on a la relation :

$$\|x\|^2 = \|p_F(x)\|^2 + d(x, F)^2$$

si F est muni d'une base orthonormale (e_1, e_2, \dots, e_p) alors on a :

$$d(x, F) = \|x\|^2 - \sum_{j=i}^p |(e_j|x)|^2$$

On trouvera des détails sur cette question dans [1] pages 28-30 et [9] pages 20-21.

4.1.2 Quelques endomorphismes importants

Nous venons de considérer les projecteurs. Le cas le plus important de projection se réalise lorsque le sous-espace F est un hyperplan de E (sous-espace de codimension 1, donc de dimension

$n-1$) ; le sous-espace F^\perp est alors une droite, dirigée par un vecteur w . Le projecteur orthogonal sur la droite dirigée par w est facile à calculer d'après le théorème ; il vaut $p_w(x) = \frac{(w|x)}{\|w\|^2} w$ (nous n'avons pas supposé w unitaire ici, il faut donc le normer pour pouvoir calculer la projection). Matriciellement parlant, si nous représentons w par une matrice unicolonne W et x par X , on a

$$\|w\|^2 = X^* X \text{ et } (w|x) = W^* X$$

d'où :

$$p_w(x) = \frac{W^* X}{W^* W} W = W \frac{W^* X}{W^* W} = \frac{1}{W^* W} (WW^*) X$$

ce qui prouve que la matrice de p_w est $\frac{1}{W^* W} (WW^*)$

Dès lors, le projecteur orthogonal sur $F = \{w\}^\perp$ est $p_F = \text{Id} - p_w$, et se calcule par :

$$p_F(x) = x - \frac{(w|x)}{\|w\|^2} w$$

et sa matrice est :

$$P_F = I_n - \frac{1}{W^* W} (WW^*)$$

Plus généralement, si F est un sous-espace de dimension p possédant une base orthonormale (w_1, \dots, w_p) , on peut représenter ces vecteurs par des matrices unicolones W_1, \dots, W_p , et les rassembler dans une matrice W . La matrice de p_F est alors WW^* . En effet, on a bien :

$$WW^* W_i = W \begin{pmatrix} W_1^* \\ \vdots \\ W_i^* \\ \vdots \\ W_p^* \end{pmatrix} W_i = W \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = W_i$$

et, par la même méthode, si X est orthogonal aux W_i alors :

$$WW^* X = 0$$

À la projection orthogonale on peut associer une transformation très importante de l'espace, la symétrie orthogonale (elle sert dans la méthode de Householder).

Définition 14. On appelle **symétrie orthogonale** par rapport à un sous-espace F de E l'endomorphisme de E :

$$s_F = 2p_F - \text{Id}$$

qui laisse les vecteurs de F invariants et change ceux de F^\perp en leurs opposés. On a $s_F^2 = \text{Id}$.

Dans le contexte matriciel, si F est un hyperplan et que w est un vecteur orthogonal à F , on a la matrice

$$(H) \quad H_w = I_n - \frac{2}{W^* W} (WW^*)$$

Réiproquement, on vérifie sans peine que toute matrice de cette forme est une matrice de symétrie orthogonale.

Les symétries orthogonales ont l'importante qualité d'être des automorphismes orthogonaux (des isométries) ; en effet, on a

$$\begin{aligned} \|s_F(x)\|^2 &= \|p_F(x) - p_{F^\perp}(x)\|^2 = \|p_F(x)\|^2 + \|p_{F^\perp}(x)\|^2 \\ &= \|p_F(x) + p_{F^\perp}(x)\|^2 = \|x\|^2 \end{aligned}$$

par le théorème de Pythagore.

Enfin, nous pouvons introduire un troisième type d'endomorphisme, les rotations (dans le seul cas d'un espace E euclidien réel).

Définition 15. Soit F un sous-espace de E de codimension 2 (donc de dimension $n - 2$). On appelle **rotation** autour de F et d'angle θ l'endomorphisme de E dont la restriction à F est l'identité et la restriction au plan F^\perp est une rotation d'angle θ .

Les rotations sont encore des automorphismes orthogonaux.

Sur le plan matriciel, nous n'aurons qu'à considérer des rotations par rapport à des sous-espaces bien particuliers. Soit une base orthonormale (e_1, \dots, e_n) de E , et soient i et j deux indices compris entre 1 et n . Considérons le sous-espace F tel que F^\perp soit engendré par les vecteurs e_i et e_j . La matrice de la rotation d'angle θ , par rapport à F , est

$$P_{ij}(\theta) = \begin{pmatrix} 1 & i \downarrow & 0 & j \downarrow & 0 \\ & \ddots & & & \\ & 1 & \cos \theta & -\sin \theta & \\ & & 1 & 0 & \\ 0 & & 0 & 1 & 0 \\ & j \rightarrow & & & \\ & & \sin \theta & \cos \theta & \\ 0 & 0 & 1 & & \\ & & & & 1 \end{pmatrix}$$

On peut noter que l'angle d'une rotation n'est connu qu'au signe près (on n'a pas d'orientation prédefinie dans F^\perp) et se calcule donc par son cosinus, donc par la trace de la matrice qui vaut $n - 2 + 2 \cos \theta$.

4.2 Procédés d'orthogonalisation

Considérons un système libre (linéairement indépendant) de vecteurs (v_1, \dots, v_p) de E . Il est souvent utile de remplacer un tel système par un système orthonormal qui engendre le même sous-espace. Nous allons confirmer la possibilité de cette opération puis étudier des algorithmes qui la réalisent effectivement.

4.2.1 Théorie du procédé de Schmidt

Théorème 4. Pour tout système libre (v_1, \dots, v_p) de vecteurs de E il existe un unique système de vecteurs (u_1, \dots, u_p) ayant les propriétés suivantes :

- (i) (u_1, \dots, u_p) est orthonormal ;
- (ii) pour tout $i \leq p$ le sous-espace F_i engendré par (v_1, \dots, v_i) coïncide avec le sous-espace engendré par (u_1, \dots, u_i) ;
- (iii) pour tout $i \leq p$ on a $(u_i | v_i) > 0$.

Preuve. \diamond Nous ne montrerons que l'existence. On procède par récurrence sur p .

Pour $p = 1$ la réponse est immédiate, c'est $u_1 = \frac{1}{\|v_1\|} v_1$.

Admettons que le procédé de Schmidt soit réalisable pour $p - 1$ vecteurs, et appliquons-le au système (v_1, \dots, v_{p-1}) , obtenant un sys-

tème (e_1, \dots, e_{p-1}) . Considérons la projection orthogonale w de v_p sur F_{p-1} . Selon l'hypothèse de récurrence F_{p-1} est engendré par (u_1, \dots, u_{p-1}) , et le théorème de la projection orthogonale montre que

$w = \sum_{i=1}^{p-1} (u_i | v_p) u_i$. Dans ces conditions, le vecteur $v_p - w$ est

orthogonal à F_{p-1} donc à u_1, \dots, u_{p-1} . Il suffira de le normer pour obtenir le vecteur u_p cherché. La norme de $v_p - w$ se calcule par le théorème de Pythagore et vaut $\sqrt{\|v_p\|^2 - \|w\|^2}$, ce qui donne :

$$u_p = K \left(v_p - \sum_{i=1}^{p-1} (u_i | v_p) u_i \right) \text{ avec } K = \left(\|v_p\|^2 - \sum_{i=1}^{p-1} |(u_i | w)|^2 \right)^{-1/2}$$

et assure que u_p appartient à F_p . Le système (u_1, \dots, u_p) est orthonormal, donc libre ce qui assure la propriété (ii) par la dimension (elle vaut p). Enfin, on a :

$$(v_p | u_p) = K(v_p | w) = K(v_p - w | w) + K\|w\|^2 = K\|w\|^2 > 0 \diamond$$

Le procédé de Schmidt présente un grand intérêt théorique (on peut même le mettre en œuvre sur un système lié ; il s'arrête alors pour trouver le rang du système et une relation de dépendance linéaire). Cependant, il n'est numériquement ni commode ni stable (le fait de réutiliser les résultats des calculs antérieurs a tendance à augmenter les erreurs d'arrondi). Matriciellement, il permet de factoriser les matrices (nous nous limiterons au cas $p = n$) :

Théorème 5. Soit A une matrice inversible de taille n . Il existe une matrice orthogonale Q et une matrice triangulaire supérieure R telles que $A = QR$.

Preuve. \diamond Considérons les colonnes de A comme un système de n vecteurs $A = (v_1, \dots, v_n)$ décomposés dans une base $\varepsilon = (e_1, \dots, e_n)$.

L'orthonormalisation de ces vecteurs conduit à une base orthonormale $U = (u_1, \dots, u_n)$.

La matrice Q ayant pour colonnes les composantes des u_i est bien une matrice orthogonale.

Chacune de ces deux matrices peut être considérée comme matrice de passage de la base de référence ε à une base différente. Suivant la même idée, la matrice U de passage de la base A à la base U est triangulaire supérieure en raison du procédé de Schmidt lui-même. Enfin, les matrices de passage se composent pour donner $AU = Q$, car si un même vecteur x (quelconque) est représenté par X_0 dans ε , X dans A et X' dans U , on a :

$$X_0 = AX = A(UX') = QX'$$

l'inverse de R de U est aussi triangulaire supérieure et donne bien $A = QR$. \diamond

4.2.2 Méthode de Householder

Au lieu de calculer de proche en proche, on va procéder plus globalement (toujours sur une matrice A inversible).

Lemme 2. Soit un vecteur a non nul de E identifié à \mathbb{K}^n par le choix d'une base orthonormale (e_1, \dots, e_n) . On pose $a = (a_1, \dots, a_n)$.

Soit $w = a - \|a\|e_1$; on a alors

$$H_w a = \|a\|e_1$$

H_w désignant la symétrie orthogonale qui change w en $-w$ et laisse l'orthogonal de w invariant.

Preuve. \diamond On a par définition $H_w(w) = -w$ et $H_w(u) = u$ si u est orthogonale à w ; ce qui est le cas de $a + \|a\|e_1$ (les diagonales du losange ayant pour sommets 0 , a , $\|a\|e_1$ et $a + \|a\|e_1$ sont en effet perpendiculaires).

Soit :

$$H_w(a - \|a\|e_1) = -a + \|a\|e_1$$

et

$$H_w(a + \|a\|e_1) = a + \|a\|e_1$$

donc, par addition : $H_w(a) = \|a\|e_1$

En fait, H_w échange les vecteurs a et $\|a\|e_1$, et c'est la symétrie par rapport à l'hyperplan médiateur de a et $\|a\|e_1$. \diamond

Notons qu'on a calculé la matrice de H_w ci-dessus ; voir (H). Le calcul des images de différents vecteurs par H_w n'est pas difficile, car il ne requiert aucune racine carrée ; et de plus se fait par :

$$H_w(x) = X - \frac{2}{W^* W} W(W^* X) = X - \frac{2}{W^* W} (W^* X) W$$

Le facteur $\frac{2}{W^* W}$ ne dépend que de w et peut être calculé une fois pour toutes. Le facteur $W^* X$ est simplement le produit scalaire de w et de x ; il ne nécessite pas beaucoup de calculs (environ n). Ainsi, le calcul de $H_w(x)$ consomme environ $2n$ multiplications et peut être considéré comme efficace.

Soit alors une matrice carrée réelle $A = (a_{ij})$. Soit a la première colonne de A ; si a est nul, on ne fait rien ; sinon, en calculant w comme ci-dessus on obtient une matrice $H_w A$ dont la première colonne est $\|a\|e_1$, c'est-à-dire ayant des coefficients tous nuls sauf sur la diagonale où apparaît un coefficient positif. Notons que si a est un vecteur de petite norme le calcul risque d'être fort peu précis ; peut-être parviendra-t-on à stabiliser l'opération en travaillant sur une autre colonne, mais ce n'est pas évident.

Cela fait, on peut recommencer l'opération dans les mêmes conditions sur la sous-matrice de A formée des lignes et colonnes numérotées entre 2 et n , sans toucher à la première ligne. Comme A est inversible, on n'aura (en théorie) jamais de coefficient nul sur la diagonale et on pourra poursuivre jusqu'au bout. Les combinaisons linéaires sur les lignes de A qui correspondent aux différents produits matriciels par des matrices de symétrie sont prescrites à chaque étape.

À la fin du procédé ($n - 1$ étapes), on aura multiplié A à gauche par $n - 1$ matrices de symétries orthogonales (soit S leur produit : c'est une matrice orthogonale) et obtenu une matrice triangulaire supérieure, ayant de surcroît ses coefficients diagonaux strictement positifs. C'est exactement la matrice R attendue. Pour calculer, s'il est besoin, la matrice Q telle que $A = QR$, il suffit d'appliquer les mêmes formules à une matrice agrandie $A' = (A | I_n)$, ce qui va conduire à $SA' = (R | S)$; on calcule très facilement $Q = S^{-1} = S^*$, parce que S est orthogonale.

4.2.3 Méthode de Givens

On peut aussi réaliser l'orthonormalisation en utilisant des matrices de rotation au lieu de matrices de symétrie. Montrons qu'on peut choisir θ tel que la matrice $P_{ij}(\theta)$ transforme un vecteur x non nul, de composantes x_i , en un vecteur x' tel que $x'_2 = 0$. Pour cela il suffit de trouver un indice i tel que x_i soit non nul (il serait sage de prendre i tel que $|x_i|$ soit maximal), et de faire agir $P_{2i}(\theta)$; on calcule les seules composantes de $x' = P_{2i}(\theta)x$ distinctes de celles de x :

$$x'_2 = \cos \theta x_2 - \sin \theta x_i, \quad x'_i = \sin \theta x_2 + \cos \theta x_i$$

En annulant x'_2 on impose la relation $\cos^2 \theta x_2^2 = \sin^2 \theta x_i^2$ d'où il résulte que $\cos^2 \theta (x_2^2 + x_i^2) = x_i^2$.

On peut choisir $\cos \theta = \frac{x_i}{\sqrt{x_2^2 + x_i^2}}$ qui entraîne

$$|\sin \theta| = \frac{x_2}{\sqrt{x_2^2 + x_i^2}}.$$

A priori, θ était choisi entre 0 et π , mais si on change éventuellement θ en $-\theta$ il vient $\sin \theta = \frac{x_2}{\sqrt{x_2^2 + x_i^2}}$. Le calcul de θ n'est pas réellement nécessaire, puisque l'application de P_{2i} sur une matrice A ne requiert que $\cos \theta$ et $\sin \theta$; il n'y a donc qu'une racine carrée à calculer. De plus, il est possible de reformuler une expression comme

$\frac{x_2}{\sqrt{x_2^2 + x_i^2}}$ pour ne pas courir le risque d'une division de deux nombres très petits ; il suffit de mettre en facteur la composante de plus grand module, par exemple x_i :

$$\frac{x_2}{\sqrt{x_2^2 + x_i^2}} = \frac{t}{\sqrt{1 + t^2}} \text{ avec } t = \frac{x_2}{x_i}$$

Une matrice A inversible étant donnée, considérons donc le vecteur x constituant la première colonne de A , et appliquons-lui la transformation (de Givens) précédente, qui revient à multiplier à gauche par une matrice $P_{2i}(\theta)$.

Posons $A' = P_{2i}(\theta) = (a'_{ij})$; elle vérifie $a'_{21} = 0$. Les coefficients de A' se calculent à partir de ceux de A par des expressions n'utilisant que des additions, multiplications, divisions et racines carrées ; seules les lignes de numéros 2 et i de A' diffèrent de celles de A , de sorte qu'il y a environ $2n$ additions et $4n$ multiplications à faire pour calculer A' .

On peut ensuite recommencer ce travail pour annuler a'_{31} , multipliant A' par une matrice $P_{3j}(\theta_3)$. Dans cette action, le coefficient nul a'_{21} ne sera pas perdu, parce que j ne peut pas être égal à 2 , et que $P_{3j}(\theta_3)$ n'agit pas sur la ligne 2 . Ainsi, il est possible d'annuler successivement tous les coefficients de la première colonne de A , à part le coefficient diagonal. Il suffit alors de recommencer sur la sous-matrice des lignes et colonnes numérotées de 2 à n pour amener A à une forme triangulaire. Le résultat est équivalent à la factorisation QR , à ceci près que la matrice Q obtenue ici est orthogonale directe et que R n'a pas nécessairement ses coefficients diagonaux positifs. On peut évidemment remédier à ce fait en multipliant quelques lignes par -1 .

La méthode de Givens, qui n'est pas plus coûteuse que la méthode Householder, présente quelques avantages sur cette dernière, et quelques inconvénients aussi. La nécessité de calculer des racines carrées peut être gênante ; mais il est possible de modifier l'algorithme pour ne pas utiliser de racine (voir [1]). L'intérêt de cette méthode est de savoir exactement quel est le travail à accomplir pour annuler un coefficient précis de A ; si A est une matrice « creuse » cette considération peut éviter beaucoup de calculs inutiles.

4.2.4 Applications

L'orthonormalisation (calcul de Q et R telles que $A = QR$) permet, entre autres, de résoudre des équations linéaires de manière stable et efficace. En effet, si on se propose de résoudre une équation linéaire $Ax = b$, il suffit de calculer $b' = Qb$ (au cours de l'algorithme

de Householder ou de Givens cela se fait aisément, avec les mêmes formules de transformation, ce qui revient à agrandir A d'une colonne supplémentaire), puis de résoudre le système triangulaire $Rx = c$.

4.3 Méthode de Cholesky

Nous avons vu à propos de la méthode du pivot la factorisation LU (méthode de Crout). Dans le cas d'une matrice A symétrique (ou hermitienne) définie positive, on peut se demander s'il est possible de donner une factorisation symétrique. Commençons par quelques définitions et rappels.

Définition 16. Une matrice A symétrique réelle est dite **définie positive** si la forme quadratique $Q_A(X) = X^* AX$ (agissant sur \mathbb{R}^n) est définie positive, c'est-à-dire qu'elle ne prend que des valeurs strictement positives à l'exception de $Q_A(0) = 0$.

Une matrice A hermitienne complexe est dite **définie positive** si la forme hermitienne $H_A(X) = X^* AX$ (agissant sur \mathbb{C}^n) est définie positive, c'est-à-dire qu'elle ne prend que des valeurs (réelles) strictement positives à l'exception de $H_A(0) = 0$.

Proposition 18.

Une matrice définie positive a ses valeurs propres et son déterminant tous strictement positifs.

Preuve. \diamond Si A est une matrice hermitienne définie positive et si V en est un vecteur propre, alors :

$$AV = \mu V \text{ et } V^* AV = \mu \|V\|^2 > 0$$

(c'est bien un nombre réel). Et le déterminant est égal au produit des valeurs propres (éventuellement répétées), donc il est réel et strictement positif. \diamond

Il est aisément de construire des matrices définies positives : il suffit de choisir une matrice inversible M et de poser $A = M^* M$. En effet, on a :

$$X^* M^* MX = (MX)^* (MX) = \|MX\|^2 \geq 0$$

pour tout vecteur X de \mathbb{K}^n , et le résultat est nul si, et seulement si, X est nul par inversibilité de M . La méthode de Cholesky vise, précisément, à obtenir une telle factorisation pour toute matrice A définie positive.

Théorème 6. Une matrice A carrée symétrique ou hermitienne est définie positive si, et seulement si, il existe une matrice triangulaire supérieure T , inversible, telle que $A = T^* T$. Si on impose, de plus, à T d'avoir des coefficients diagonaux strictement positifs, alors elle est unique.

Preuve. \diamond La nécessité a déjà été montrée.

L'unicité se traite ainsi : si $T^* T = S^* S$, T et S ayant les caractéristiques voulues, on a :

$$S^{*-1} T^* TS^{-1} = (TS^{-1})^* TS^{-1} = I$$

donc TS^{-1} est une matrice orthogonale réelle ou unitaire complexe et, en même temps, triangulaire supérieure avec une diagonale strictement positive. En écrivant l'orthogonalité des colonnes de cette matrice, de proche en proche, on voit qu'elle est diagonale ; ses coefficients diagonaux doivent être de module 1 et étant positifs ils valent 1 ; d'où $T = S$.

En ce qui concerne l'existence, on peut considérer la forme hermitienne (ou quadratique, dans le cas réel) associée à A qui s'écrit

$H_A(x) = H^* AX$ pour $x \in \mathbb{K}^n$. Étant définie positive, elle définit un produit scalaire sur \mathbb{K}^n , de la forme $[x, y] = X^* AY$ (qui n'est pas le produit scalaire usuel). Appliquons le procédé de Schmidt (ou toute méthode d'orthonormalisation) dans ce contexte, sur la base canonique (e_i) de \mathbb{K}^n , pour obtenir une base (z_i) orthonormée pour H_A . Un vecteur x de \mathbb{K}^n est représenté par une matrice unicolumnne X_0 dans la base (e_i) et par X dans (z_i) ; la matrice de passage S est triangulaire supérieure, comme l'impose le procédé de Schmidt. On calcule alors $[x, y]$ de deux manières :

— d'une part, c'est :

$$X_0^* A Y_0 = (SX)^* ASY = X^* S^* ASY$$

— d'autre part, comme la base résultante est orthonormée, on a :

$$[x, y] = X^* Y$$

L'identité de ces deux expressions (pour tous x et y) amène $S^* AS = I_n$, soit $A = T^* T$ en posant $T = S^{-1}$. \diamond

■ Examinons maintenant l'algorithme qui calcule T . Notons $A = (a_{ij})$ et $T = (t_{ij})$. On connaît tout de suite t_{11} par la relation $\bar{t}_{11} t_{11} = a_{11}$; le choix de t_{11} est possible parce que :

$$a_{11} = E_1^* AE_1 > 0$$

E_1 désignant la matrice unicolumnne associée au premier vecteur de la base canonique de \mathbb{R}^n ou \mathbb{C}^n .

■ On peut maintenant écrire $a_{1i} = t_{11} t_{1i}$ qui fixe aussitôt $t_{1i} = \frac{a_{1i}}{t_{11}}$ pour tout $i > 1$. Cela fait, on peut passer à :

$$a_{22} = |t_{22}|^2 + |t_{12}|^2 = t_{22}^2 + \frac{|a_{12}|^2}{a_{11}}$$

soit :

$$t_{22}^2 = a_{22} - \frac{a_{12} a_{21}}{a_{11}} = \frac{\det A_2}{\det A_1}$$

notant A_k la sous-matrice principale de A , formée des k premières lignes et colonnes. La sous-matrice A_k est définie positive, car elle correspond à la restriction de la forme quadratique (ou hermitienne) $H_A(X) = X^* AX$ au sous-espace formé des vecteurs ayant leurs $n-k$ dernières composantes nulles. Par conséquent, elle a un déterminant strictement positif, ce qui confirme la possibilité de choisir t_{22} (strictement positif).

On continue de même. Si les $k-1$ premières lignes de T ont été déterminées, pour trouver la k -ième on commence par t_{kk} qui est déterminé par :

$$a_{kk} = t_{kk}^2 + \sum_{i=1}^{k-1} |t_{ik}|^2, \text{ soit } t_{kk}^2 = \sqrt{a_{kk} - \sum_{i=1}^{k-1} |t_{ik}|^2}$$

Puis on trouve pour $i > k$:

$$a_{ki} = t_{kk} t_{ki} + \sum_{j=1}^{k-1} \bar{t}_{jk} t_{ji}, \text{ soit } t_{ki} = \frac{a_{ki} - \sum_{j=1}^{k-1} \bar{t}_{jk} t_{ji}}{t_{kk}}$$

■ Notons qu'il y a une autre version de cet algorithme, échangeant le rôle des lignes et des colonnes. D'autre part, on peut utiliser la méthode de Cholesky pour vérifier qu'une matrice est définie positive : si elle ne l'est pas, l'algorithme n'aboutit pas (l'un des t_{kk} ne peut plus être choisi).

■ En conclusion, la méthode de Cholesky, qui a l'avantage d'être numériquement assez stable, permet la résolution d'un système

linéaire $AX = B$ en se ramenant à la résolution de deux systèmes triangulaires :

$$T^* Y = B \text{ et } TX = Y$$

Cela est surtout intéressant lorsqu'on doit résoudre de nombreux systèmes avec la même matrice A mais où seul le second membre B change.

4.4 Décomposition en valeurs singulières

Nous aurons besoin du résultat suivant, concernant un endomorphisme u autoadjoint de E (c'est-à-dire, tel que $(x|u(y)) = (u(x)|y)$ pour tous vecteurs x et y de E).

Théorème spectral. Étant donné un endomorphisme autoadjoint u d'un espace euclidien ou hermitien E (de dimension finie), les sous-espaces propres de u sont deux à deux orthogonaux et leur somme est égal à l'espace E ; en particulier, u est diagonalisable dans une base orthonormale.

Étant donnée une matrice A symétrique réelle (resp. hermitienne complexe), il existe un matrice orthogonale (resp. unitaire) U telle que $U^* AU$ soit diagonale.

Le théorème spectral permet de trouver de nombreuses factorisations matricielles, dont la décomposition en valeurs singulières est le meilleur exemple.

Théorème 7. Soit A une matrice quelconque à coefficients réels ou complexes (ayant n lignes et p colonnes). Il existe deux matrices orthogonales ou unitaires U et V telle que $U^* AV$ soit diagonale (c'est-à-dire, ait tous ses coefficients nuls hors de la diagonale) et ait des coefficients diagonaux positifs ou nuls.

Preuve. ◊ Considérons $A^* A$, qui est une matrice hermitienne positive comme on l'a vu à propos de la méthode de Cholesky. Elle ne sera définie positive que si A est inversible, ce qui nécessite qu'elle soit carrée. Cependant, on peut diagonaliser une telle matrice d'après le théorème spectral. Il vient donc :

$$A^* A = V D V^*$$

V étant une matrice unitaire et la matrice D diagonale, ayant comme coefficients diagonaux les valeurs propres de $A^* A$. Comme cette dernière est positive, ses valeurs propres le sont. On peut donc écrire $D = E^2$, en prenant pour E une matrice diagonale ayant pour coefficients les racines carrées des coefficients de D . On a la relation :

$$V^* A^* AV = (AV)^* (AV) = D$$

diagonale de coefficients μ_i^2 .

Soient W_1, \dots, W_p les colonnes de AV (ce sont les vecteurs \mathbb{K}^n). La formule précédente exprime que :

$$(W_i | W_j) = 0 \text{ si } i \neq j \text{ et } \|W_i\|^2 = \mu_i^2$$

Nous pouvons supposer sans restriction que μ_i est non nul pour $i \leq r$ et que μ_i est nul sinon. Nous pouvons alors introduire

$U_i = \frac{1}{\mu_i} W_i$ pour $i \leq r$; ils forment un système orthonormal de

\mathbb{K}^n , et une base de l'image de A . On peut les compléter de manière assez arbitraire en prenant une base orthonormale du sous-espace polaire (orthogonal) de l'image de A dans \mathbb{K}^n ; pour obtenir une

base orthonormale (U_i) de cet espace. Bien entendu, si $r = n$ il n'y a rien à faire.

Soit U la matrice ayant pour colonnes les vecteurs U_i ; elle est unitaire et on a, pour $i \leq r$:

$$U_i^* A V_i = \frac{1}{\mu_i} W_i^* W_i = \frac{\mu_i^2}{\mu_i} = \mu_i$$

pour $i > r$ le produit $U_i^* A V_i$ est nul de même que $U_i^* A V_j = 0$ pour $i \neq j$.

Dès lors, $U^* A V$ est la matrice diagonale de coefficients μ_i . ◊

Définition 17. On appelle **valeurs singulières** de A les coefficients diagonaux de la matrice $U^* A V$ introduite ci-dessus ; ce sont les valeurs propres de $A^* A$. Le nombre de valeurs singulières non nulles est exactement égal au rang de A .

Le calcul de la décomposition en valeurs singulières d'une matrice n'est pas une chose facile, car il repose sur la diagonalisation de la matrice hermitienne $A^* A$. Pour ce faire, on dispose de méthodes numériques relativement efficaces comme les méthodes itératives de Jacobi ou du QR. Le lecteur pourra consulter les ouvrages spécialisés à ce sujet, notamment [1], [9] pages 293 et [13] page 67.

4.5 Pseudo-inverses

Au cours du XX^e siècle, les mathématiciens Moore et Penrose ont introduit une extension commode de la notion de matrice inverse dans un contexte euclidien ou hermitien, qui permet une bonne prise en compte des notions de solutions approchées d'une équation linéaire. Cet outil sert dans les problèmes de traitement du signal, réseaux de neurones, etc.

4.5.1 Définition

Définition 18. Soit A une matrice quelconque à coefficients réels ou complexes (ayant n lignes et p colonnes). On appelle **pseudo-inverse** de A une matrice B (ayant p lignes et n colonnes) telle que les relations suivantes soit satisfaites :

$$ABA = A \quad BAB = B \quad (AB)^* = AB \quad (BA)^* = BA$$

Dans le cas des matrices carrées inversibles, cette définition n'a guère d'intérêt.

Proposition 19.

Si A est une matrice inversible alors elle admet pour pseudo-inverse son inverse habituelle.

Théorème 8. La pseudo-inverse d'une matrice (réelle ou complexe) A existe et est unique.

Preuve. ◊ Notons provisoirement B une éventuelle pseudo-inverse de A . Notons que si B existe alors :

$$(AB)^2 = AB = (AB)^*$$

ce qui donne à AB le caractère d'un projecteur orthogonal. Son image est a priori incluse dans l'image de A , mais la formule $ABA = A$ montre l'identité des deux images.

De même, son noyau est celui de B qui est nécessairement égal au sous-espace polaire de l'image de A .

Par symétrie, BA est nécessairement un projecteur orthogonal d'image l'image de B et de noyau le noyau de A ; notamment, BA doit agir comme l'identité sur le sous-espace polaire du noyau de A . Or, la restriction de A à un supplémentaire de son noyau est un isomorphisme vers l'image de A (voir l'article AF 85, paragraphe 1.3.2, proposition 12). Ainsi, B agit sur l'image de A comme la réciproque de l'isomorphisme évoqué, et est nulle sur le sous-espace polaire (ou orthogonal) de l'image de A , ce qui assure l'unicité.

En ce qui concerne l'existence, elle découle facilement de la caractérisation géométrique qui vient d'être donnée. ◇

Notation : on posera désormais $B = A^+$.

Proposition 20.

Si $A = U\Delta V^$ est la décomposition en valeurs singulières de A , les valeurs singulières non nulles de A étant μ_1, \dots, μ_r , alors :*

$$A^+ = V\Delta^+U^*$$

et les valeurs singulières de A^+ , figurant sur la diagonale de Δ^+ , sont

$$\frac{1}{\mu_1}, \dots, \frac{1}{\mu_r} \text{ et } 0.$$

Preuve. ◇ Il suffit d'introduire la matrice Δ^+ remplie comme indiqué, et de vérifier que $B = V\Delta^+U^*$ vérifie les quatres axiomes de définition de la pseudo-inverse. Ces calculs sont faciles (par exemple :

$$BA = V\Delta^+U^*$$

et

$$BAB = V\Delta^+\Delta^+U^* = B$$

parce que $\Delta^+\Delta^+$ coïncide avec Δ^+). ◇

Proposition 21.

Soit B un vecteur de \mathbb{K}^n . Le vecteur $X_0 = A^+B$ est l'unique vecteur minimisant la norme $\|AX - B\|$, lorsque X décrit \mathbb{K}^p .

Preuve. ◇ Dans cet énoncé, précisons bien que la norme utilisée est la norme euclidienne canonique de l'espace \mathbb{K}^n . Nous avons $AX_0 = AA^+B$ et nous savons que AA^+ est le projecteur orthogonal sur l'image de A ; par conséquent, AA^+B est le vecteur de l'image de A le plus proche possible de B (voir le théorème de la projection orthogonale, § 4.1.1), donc :

$$\|AA^+B - B\| = \min_{Y \in \text{Im}(A)} \|Y - B\| = \min_X \|AX - B\|$$

ce qu'il fallait. ◇

Définition 19. Dans les conditions de la proposition précédente, X_0 est appelé la **pseudo-solution** (au sens des moindres carrés) de l'équation linéaire $AX = B$.

4.5.2 Calcul pratique

Les pseudo-inverses n'ont d'intérêt que pour des matrices singulières, puisque, pour les matrices inversibles, on a $A^+ = A^{-1}$. Insistons sur le fait qu'il n'est pas nécessaire de supposer que A est une matrice carrée. Le problème du calcul effectif est cependant délicat pour les matrices singulières, puisque toute erreur d'arrondi détruit (en principe) le rang de la matrice.

Voici l'algorithme de Kalaba-Rashkov, qui est assez stable pour déterminer la pseudo-inverse ; cette stabilité a un prix, comme on le verra.

Proposition 22.

$$\text{On a } A^+ = \lim_{t \rightarrow 0} (A^* A + tI_n)^{-1} A^*.$$

Preuve. ◇ Pour les vecteurs du sous-espace polaire de l'image de A (donc du noyau de A^+ , égal au noyau de A^*) la formule se résume à $0 = 0$. Pour ceux de l'image de A , on calcule $(A^* A + tI_n)^{-1} A^* A$ grâce à la décomposition en valeurs singulières de A :

$$A = UDV^*$$

$$A^* A = VD^* DV$$

et

$$(A^* A + tI_n)^{-1} A^* A = V(D^* D + tI_n)^{-1} D^* DV^*$$

et la matrice $(D^* D + tI_n)^{-1} D^* D$ a pour coefficients diagonaux $\frac{\mu_i}{\mu_i + t}$ ce qui tend vers 1 si $\mu_i \neq 0$ et 0 sinon, résultat qui est en accord avec la pseudo-inverse. ◇

Cette formule est tout à fait impraticable pour le calcul numérique !

Éliminons désormais les cas où A est nulle ou inversible.

On écrit alors :

$$\Delta = \det(A^* A + tI_p) = t^p + \delta_1 t^{p-1} + \dots + \delta_p$$

c'est à quelques signes près le polynôme caractéristique de $A^* A$, qui possède donc toutes ses racines dans \mathbb{R} et 0 comme racine de multiplicité $p - r$ si r est le rang de A . On a donc en fait $\delta_j = 0$ pour $j < r$ et $\delta_r \neq 0$.

On peut, de même, considérer la transposée de la comatrice B de $A^* A + tI_p$, qui va se développer sous une forme :

$$B = B_1 t^{p-1} + \dots + B_p$$

les B_j étant des matrices carrées et $B_1 = I_p$.

En développant le produit $B(A^* A + tI_p) = \Delta I_p = \Delta I_p$, on trouve la relation :

$$B_{j+1} = \delta_j I_p - B_j A^* A$$

Enfin, en dérivant par rapport à t le déterminant Δ on trouve que :

$$\delta_j = \frac{1}{j} \text{Tr}(B_j A^* A)$$

Dans ces conditions, $(A^* A + tI_p)^{-1} A^* = \frac{1}{\Delta} BA^*$; lorsque t tend

vers 0 le dénominateur Δ est équivalent à $\delta_r t^{p-r}$; comme la limite doit être finie et non nulle, au numérateur on doit avoir $B_j = 0$ pour $j < r$ et $B_r \neq 0$, et $A^+ = \frac{1}{\delta_r} B_r A^*$. Pour calculer A^+ , il convient donc de calculer la suite de matrices (B_j) avec l'algorithme suivant.

Initialisations :

B_1 est une matrice identité de taille p

$C = A^* A$

$j = 1$

$r = \text{rg}(A)$

Itération :

Faire varier j de 1 à r et calculer $D = B_j C$ puis

$\delta_j = \frac{1}{j} \text{Tr}(D)$ et $B_{j+1} = \delta_j I_p - D$ (on peut recouvrir B_j par B_{j+1}).

Fin : $A^+ = \frac{1}{\delta_r} B_r A^*$

Cet algorithme consomme environ rn^3 calculs.

Dans l'algorithme précédent, le critère d'arrêt des itérations est déduit de la connaissance (supposée) du rang de la matrice. Si ce rang n'est pas connu à l'avance, on peut songer à modifier le test et à s'arrêter dès qu'une matrice D est nulle. Évidemment, les erreurs d'arrondi empêcheraient, dans un tel contexte, d'avoir un critère fiable. Répétons encore que, dans un problème de matrices singulières, les valeurs approchées retirent toute signification à la pseudo-inverse.

La pseudo-inverse n'est d'ailleurs pas une fonction continue de A , puisque, par exemple, elle n'est pas bornée au voisinage de la matrice nulle ; ainsi, la pseudo-inverse de la matrice aI_n est $\frac{1}{a}I_n$ et, si a tend vers 0, la pseudo-inverse en question devient infiniment grande, alors que la pseudo-inverse de la matrice nulle est elle-même.

5. Matrices creuses

Ce paragraphe décrit très brièvement les méthodes spécifiques pour traiter les problèmes de grande taille mais avec peu d'inconnues chaque équation.

5.1 Problèmes de grande taille en algèbre linéaire

Les problèmes obtenus par discréttisation d'équations aux dérivées partielles engendrent facilement des matrices de taille gigantesque. Par exemple, la discréttisation de l'équation de la chaleur en dimension 3 sur un parallélépipède $[0, 1]^3$ nécessite de découper le domaine suivant trois directions ; si on prend comme inconnues la température en chacun des points du cube discréttisé, on a N^3 inconnues si chaque côté du cube est discréttisé en N points. On aura aussi autant d'équations (linéaires) en écrivant les approximations discrètes du laplacien et des dérivées partielles, ainsi que les conditions aux limites. Une discréttisation avec 100 points par axe entraînera un système de 10^6 équations avec 10^6 inconnues, donc pourvu d'une matrice ayant 10^{12} coefficients. Une telle matrice n'est pas représentable dans la mémoire d'un ordinateur. Cependant, la plupart des coefficients de cette matrice sont nuls, puisque chaque équation ne comporte que 6 ou 7 termes au plus. De telles matrices sont dites « **creuses** ».

Il faut donc se pencher sur le mode de représentation des matrices creuses, et sur l'influence de ces modes de représentation sur le choix des algorithmes.

5.2 Modes de représentation d'une matrice creuse

Une matrice creuse a généralement très peu de coefficients non nuls. On peut stocker ceux-ci dans un tableau ou une liste contenant l'information sur la position et la valeur du coefficient. Cependant, une telle représentation peut être inefficace lorsqu'il s'agit de modifier quelques coefficients de la matrice. Lorsqu'il faut économiser beaucoup de place, une structure de liste chaînée peut convenir : on représente, pour chaque ligne de la matrice, les coefficients non nuls avec une information permettant de passer au coefficient suivant.

On peut aussi représenter la matrice en se basant sur la connaissance a priori des endroits où se trouvent les coefficients non nuls.

Par exemple, une **matrice symétrique ou triangulaire** ne contient de l'information que sur un triangle de coefficients, et il convient de stocker seulement les coefficients utiles, en un tableau de $\frac{n(n+1)}{2}$ valeurs.

De même, une **matrice tridiagonale** de taille n ne consomme que $3n - 2$ cases de mémoire.

Plus **généralement**, si pour une ligne donnée (numéro i) on sait que les coefficients non nuls se trouvent entre les colonnes $j_{\min}(i)$ et $j_{\max}(i)$, il suffit de garder la suite des coefficients concernés ainsi que $j_{\min}(i)$ et $j_{\max}(i)$.

5.3 Algorithmes spécifiques pour les matrices creuses

Une fois que le problème de la représentation de la matrice est résolu, il faut prendre garde au fait que le nombre de calculs à réaliser est très élevé, à cause de la taille de la matrice. Dans ces conditions, il faut prendre en considération l'accumulation des erreurs d'arrondi qui peut devenir gênante. Pour cette raison, on écarte généralement la méthode du pivot au profit de méthodes plus stables basées sur les factorisations de Cholesky ou du QR.

La méthode de Gauss-Seidel est largement employée pour résoudre les systèmes linéaires de grande taille, car, si elle converge de manière satisfaisante, il n'est pas nécessaire d'itérer un grand nombre de fois. De surcroît, elle est peu gourmande en stockage intermédiaire.

Notons enfin que l'efficacité de ces algorithmes ne se mesure plus seulement au nombre d'opérations arithmétiques à réaliser, mais aussi au nombre d'accès à la valeur des coefficients, la mémoire de stockage de ces coefficients n'étant pas nécessairement d'accès instantané. Dans ces conditions, la performance de l'algorithme peut tenir beaucoup plus à l'optimisation des séquences d'accès aux coefficients qu'au nombre de calculs effectués.

Méthodes numériques en algèbre linéaire

par **Robert CABANE**

*Ancien élève de l'École Normale Supérieure
Professeur de Mathématiques Spéciales au
Lycée Michel-Montaigne (Bordeaux)*

Pour en savoir plus

Logiciels et bibliothèques de calcul numérique

La mise en œuvre effective des méthodes décrites ci-dessus requiert des techniques informatiques fort précises que la taille de cet article ne permet pas d'aborder. De toutes manières, les programmes « prêts à l'emploi » ne sont pas toujours bien adaptés au traitement de situations réelles qui peuvent nécessiter des simplifications préalables, des estimations des erreurs tolérables, etc.

C'est donc sous toutes réserves que nous fournissons ci-dessous une liste nullement exhaustive des logiciels et bibliothèques existants. Ajoutons que ce domaine est susceptible d'évoluer rapidement sous l'effet de l'amélioration continue des performances des ordinateurs et de la découverte d'algorithmes toujours plus efficaces. Enfin, signalons que la validité de certaines « recettes numériques » est parfois contestée ; il convient donc de se renseigner amplement avant de faire confiance à un programme existant.

Voici une liste des principales bibliothèques (librairies) de calcul numérique linéaire :

1. EISPACK.

Bibliothèque de programmes en Fortran pour le calcul de valeurs propres et vecteurs propres. Contient des procédures pour l'évaluation du rayon spectral.

NEA Data Bank, BP 9,

91191 Gif-sur-Yvette, France

Le manuel est édité sous le titre :

Smith (B.T.), Boyle (J.M.), Dongarra (J.), Garbow (B.), Ikebe (Y.), Klema (V.C.) et Moler (C.B.) – *Matrix Eigensystem Routines : EISPACK Guide* (Programmes pour le calcul matriciel des éléments propres) Springer-Verlag, New York, 1976.

2. LINPACK.

Le manuel est édité sous le titre :

Bunch (J.R.), Dongarra (J.), Stewart (G.W.) et Moler (C.B.) - *LINPACK User's Guide* (Guide de l'utilisateur de LINPACK) SIAM Publications, Philadelphia, 1978.

3. IMSL.

Sixth floor, NBC Bldg,
7500 Bellaire Blvd,
Houston Texas 77036,
USA

4. NAG.

Mayfield House 256 Banbury Road,
Oxford OX27DE,
Grande-Bretagne

5. MATLAB.

La consultation d'Internet à partir du thème « numerical recipes » fournit, d'autre part, de précieux renseignements (bien que peu durables).

1. Site « Numerical Recipes » à l'Université de Harvard. Téléchargement de sources, informations sur les erreurs connues, documentation. Adresses :

<http://nr.harvard.edu/nr>

<http://cfatab.harvard.edu/nr>

2. Site de l'Université de Lund (Suède). Téléchargement d'algorithmes. Adresse:

<http://www.astro.lu.se/Numrecipes>

Références bibliographiques

- [1] LASCAUX (P.) et THÉODOR (R.). – *Analyse numérique matricielle appliquée à l'art de l'ingénieur*. Masson, Paris. 1993.
- [2] VOIEVODINE (V.). – *Principes numériques d'Algèbre Linéaire*. Éditions Mir, Moscou. 1980.
- [3] SIBONY (M.) et MARDON (J.-CL.). – *Systèmes linéaires et non linéaires*. Hermann, Paris. 1984.
- [4] CARNAHAN (B.), LUTHER (H.A.) et WILKES (O.J.). – *Applied Numerical Methods* (Méthodes numériques appliquées). Wiley, New York. 1969.
- [5] DEMIDOVITCH (B.) et MARON (I.). – *Éléments de calcul numérique*. Éditions Mir, Moscou. 1979
- [6] CIARLET (P.G.). – *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*. Masson, Paris. 1982.
- [7] CHATELIN (F.). – *Valeurs propres de matrices*. Masson, Paris. 1986.
- [8] GANTMACHER (F.R.). – *Théorie des Matrices I et II*. Dunod, Paris. 1966.
- [9] GOLUB (G.H.) et VAN LOAN (C.F.). – *Matrix Computations* (Calcul matriciel). John Hopkins University Press, Baltimore. 1983.
- [10] STRANG (G.). – *Linear Algebra and its Applications*. Academic Press. 1976.
- [11] CABANE (R.) et LEBŒUF (C.). – *Cours d'Algèbre Linéaire I, Espaces vectoriels et polynômes*. Ellipses, Paris. 1987.
- [12] CABANE (R.) et LEBŒUF (C.). – *Cours d'Algèbre Linéaire II, Matrices et réduction*. Ellipses, Paris. 1990.
- [13] PRESS (W.H.), FLANNERY (B.P.), TEUKOLSKY (S.A.) et VETTERLING (W.T.). – *Numerical Recipes in C, The Art of Scientific Computing, Second Edition* (Recettes numériques en langage C, l'art du calcul scientifique, seconde édition). Cambridge University Press. 1992.
- [14] PRESS (W.H.), FLANNERY (B.P.), TEUKOLSKY (S.A.) et VETTERLING (W.T.). – *Numerical Recipes in Fortran, The Art of Scientific Computing, Second Edition* (Recettes numériques en langage Fortran, seconde édition). Cambridge University Press. 1992.

MÉTHODES NUMÉRIQUES EN ALGÈBRE LINÉAIRE

- [15] ACTON (F.S.). – *Numerical Recipes that usually Work* (Recettes numériques qui fonctionnent en général). Mathematical Association of America, Washington. 1990.
- [16] RALSTON (A.) et RABINOWITZ (P.). – *A First Course in Numerical Analysis, Second Edition* (Un cours d'Analyse numérique pour débutants, seconde édition) Mc Graw-Hill, New York. 1978.
- [17] WILKINSON (J.H.) et REINSCH (C.). – *Handbook for Automatic Computation, Linear Algebra, Volume II* (Manuel de calcul automatique, Algèbre Linéaire, second volume) Springer Verlag, New York/Heidelberg. 1971.
- [18] NAUDIN (P.) et QUITTÉ (C.). – *Algorithmique algébrique* Masson, Paris. 1992.
- [19] BAREISS (E.H.). – *Sylvester's identity and multistep integer-preserving Gaussian elimination* (Identité de Sylvester et élimination de Gauss préservant les entiers) Math. Comp. vol. 22, 1968, pp. 565-578.