NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

ISBN 0-315-56456-3

Estimation du conditionnement d'une matrice

par

Omar SLIMANI

Thèse présentée à l'école des études supérieures
de l'Université d'Ottawa pour l'obtention de la maîtrise
ès sciences en mathématiques.

5 juillet 1989
Ottawa, Ontario, Canada K1N 6N5

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

A mes parents

Remerciements

Je voudrais exprimer ici mes remerciements chaleureux à mon superviseur Rémi Vaillancourt pour avoir si patiemment dirigé ma recherche, aux étudiants G. Kilani et A. Kelm pour m'avoir aidé à linotyper ma thèse, ainsi qu'au directeur de département, W. Burgess, pour m'avoir facilité son impression.

Enfin je remercie également le gouvernement de mon pays (Algérie) pour avoir financé mes études.

Abstract

It is important in practice when solving linear systems to have an economical method for estimating the condition number $\kappa(A)$ of the matrix of coefficients. Different algorithms involving $O(n^2)$ arithmetic operations are described; they give reliable indications of the order of magnitude of $\kappa(A)$.

Numerical experiments are presented in order to illustrate and compare the practical performance of these algorithms.

Résumé

Après la résolution d'un système linéaire $Ax = b$, il est toujours important d'avoir une méthode économique pour estimer le conditionnement $\kappa(A)$ de la matrice des coefficients. Différents algorithmes utilisant un ordre de $O(n^2)$ opérations arithmétiques sont décrits.

Afin d'illustrer et de comparer la performance de ces algorithmes en pratique, certains résultats expérimentaux sont présentés.

Table de matières

Introduction

L'approximation de la solution de beaucoup de problèmes que l'on rencontre en mathématiques appliquées, en physique ou dans les sciences de l'ingénieur, conduit le plus souvent à des problèmes d'analyse numérique matricielle, et plus particulièrement à la résolution de systèmes d'équations linéaires de la forme $Ax = b$, et par conséquent au problème de l'estimation du conditionnement d'une matrice.

Nous verrons, à propos du conditionnement des problèmes, l'influence que peuvent avoir les erreurs sur les données; elles résultent de ce que les données (composantes d'une matrice A, coordonnées d'un vecteur b) peuvent souvent n'être connues qu'approximativement, comme par exemple les données expérimentales.

Même quand la précision sur les données est très bonne, de petites perturbations des données peuvent avoir une influence absolument désastreuse sur le résultat.

Ces problèmes interviennent notamment régulièrement dans l'appliquation de la méthode des éléments finis à des equations aux dérivées partielles elliptiques, à des problèmes de vibrations et d'élasticité, ou en hydrodynamique.

Même lorsque les problèmes considérés ne sont pas linéaires, il arrive très souvent qu'on les ramène par des chemins appropriés, (par exemple des itérations), à des suites de problèmes linéaires.

Dans notre étude du conditionnement d'une matrice, on s'est restreint aux cas des normes l_1, l_2 et l_∞, qui sont les plus utilisées en analyse numérique.

Afin de ne pas trop allonger l'exposé, nous avons dû renoncer à inclure la description des algorithmes et programmes correspondants aux méthodes étudiées.

Chapitre 1

Rappels sur les matrices et l'analyse numérique matricielles

1.1 Introduction

Ce chapitre a pour but de rappeler, sans démontrer, un certain nombre de résultats relatifs aux matrices, leurs normes, ainsi que certains domaines de leur application dont le plus connu est le système linéaire d'équations algébriques. La dernière section introduit la notion de conditionnement d'une matrice.

1.2 Représentation matricielle d'un opérateur linéaire
1.3 Algèbre des matrices
1.4 Normes vectorielles
1.5 Normes matricielles
1.6 Décomposition en valeurs singulières
1.7 Théorie de la perturbation
1.8 Conditionnement de matrices

1.2 Représentation matricielle d'un opérateur linéaire

Soit $V_1 = \mathbf{R}^m$ et $V_2 = \mathbf{R}^n$ deux espaces vectoriels réels.

Soit $T : V_1 \longrightarrow V_2$ un opérateur linéaire.

Il est connu que T est représentable par une matrice réelle A à n lignes et m colonnes de la manière suivante:

Tout $x \in \mathbf{R}^m$ peut s'écrire sous la forme vectorielle

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}.$$

Appliquer T sur x pour obtenir $y = T(x)$ revient à multiplier x par A et obtenir le vecteur colonne y

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix};$$

où $y_i = \sum_{i=1}^{m} a_{ik} x_k$ représente la i-ème composante de y.

1.3 Algèbre des matrices [12, p. 5]

Soit $\mathbf{R}^{n \times m}$ l'espace vectoriel de toutes les matrices réelles A à n lignes et m colonnes:

$$A \in \mathbf{R}^{n \times m} \quad \Leftrightarrow \quad A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nm} \end{bmatrix}.$$

A est dite de type (n, m) si $n \neq m$; elle est dite d'ordre n si $m = n$.

Deux sous-espaces importants sont associés à toute matrice A de type (n, m). Le sous-espace image de A defini par

$$\mathrm{Im}(A) = \{y \in \mathbf{R}^n | y = Ax \quad \text{pour un certain } x \in \mathbf{R}^m\}$$

et le kernel , ou noyau, de A par

$$\mathrm{Ker}\,(A) = \{x \in \mathbf{R}^m | \, Ax = 0\}.$$

$\mathrm{Im}(A)$ est le sous-espace vectoriel de $\mathbf{R}^n$, engendré par les colonnes de A. Le rang de A est défini par

$$\mathrm{rang}\,(A) = \dim(\mathrm{Im}(A)).$$

C'est le nombre maximal de colonnes de A qui sont linéairement indépendantes. Si $n = m$, il est bien connu que les trois propositions suivantes sont équivalentes.

(i) A est régulière

(ii) Ker $(A) = \{0\}$

(iii) rang $(A) = n$

Ci-dessous, une liste de matrices spéciales est donnée. Elle servira comme référence pour le reste de l'exposé. Une matrice A d'ordre n est dite:

Symétrique	*si*	$A^T = A$
Définie positive	*si*	$x^T Ax > 0\,,\ 0 \neq x \in \mathbf{R}^n$
Positive	*si*	$x^T Ax \geq 0\,,$ pour tout $x \in \mathbf{R}^n$
Orthogonale	*si*	$A^T A = AA^T = I$
Permutation	*si*	$A = [e_{s_1}, \ldots, e_{s_n}]$
Diagonale	*si*	$a_{ij} = 0$ pour tout $i \neq j$
Tridiagonale	*si*	$a_{ij} = 0$ pour tout i, j tels que $\lvert i - j \rvert > 1$
Triangulaire supérieure	*si*	$a_{ij} = 0$ pour tout $i > j$
Hessenberg supérieure	*si*	$a_{ij} = 0$ pour tout $i > j + 1$

1.4 Normes vectorielles [12, ch. 2]

Une norme vectorielle sur $\mathbf{R}^n$ est une application

$$f : \mathbf{R}^n \longrightarrow \mathbf{R}$$

qui vérifie les propriétés suivantes:

1. $f(x) \geq 0$ pour tout $x \in \mathbf{R}^n$ avec égalité ssi $x = 0\,,$

2. $f(x + y) \leq f(x) + f(y)$ pour tout $x, y \in \mathbf{R}^n,$

3. $f(\alpha x) = |\alpha| f(x)$ pour tout $x \in \mathbf{R}^n.$

Une telle fonction sera notée par $\|.\|$.
Une classe importante de normes est

$$\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{1/p} \quad p \geq 1\,,$$

parmi lesquelles

$$||x||_1 = (|x_1| + \ldots + |x_n|),$$
$$||x||_2 = (|x_1|^2 + \ldots + |x_n|^2)^{1/2},$$
$$||x||_\infty = \max|x_i|,$$

sont les plus importantes en analyse numérique. La norme (l_2), appelée norme euclidienne, a une importance particulière du fait qu'elle est la vraie distance dans $\mathbf{R}^n$.

Un résultat classique concernant les normes l_p est l'inégalité de Hölder qui suit:

$$|x^T y| \leq ||x||_p ||y||_q \qquad (1/p) + (1/q) = 1.$$

Un cas spécial de cette dernière est l'inégalité de Cauchy-Schwartz:

$$|x^T y| \leq ||x||_2 ||y||_2.$$

Une autre particularité de la norme euclidienne est son invariance par transformation orthogonale. Soit Q une matrice orthogonale d'ordre n, alors

$$||Qx||_2^2 = (Qx, Qx) = x^T(Q^T Q)x = ||x||^2,$$

où (x, y) dénote le produit scalaire de x et y
Toutes les normes sur $\mathbf{R}^n$ sont équivalentes. C'est-à-dire, étant donné deux normes quelconques $||.||_p$ et $||.||_q$ sur $\mathbf{R}^n$, il existe deux constantes positives m et M telles que:

$$m||x||_p \leq ||x||_q \leq M||x||_p$$

pour tout x.

1.5 Normes matricielles

De manière analogue à la définition d'une norme vectorielle sur $\mathbf{R}^n$, une norme matricielle sur $\mathbf{R}^{n\times m}$ est une fonction $f : \mathbf{R}^{n\times m} \longrightarrow \mathbf{R}$ qui vérifie les propriétés suivantes:

(a) $f(A) \geq 0$ pour tout $A \in \mathbf{R}^{n\times m}$ avec l'égalité ssi $A = 0$
(b) $f(A+B) \leq f(A) + f(B)$ pour tout $A \in \mathbf{R}^{n\times m}$ et $B \in \mathbf{R}^{n\times m}$
(c) $f(\alpha A) = |\alpha| f(A)$ pour tout $A \in \mathbf{R}^{n\times m}$ et $\alpha \in \mathbf{R}$

Les normes dont l'utilisation est très fréquente en analyse numérique sont:

la norme F (norme de Frobenius)

$$\| A \|_F = [\sum_{j=1}^{m}\sum_{i=1}^{n} |a_{ij}|^2]^{1/2}, \tag{1.1}$$

(dans certains livres, elle est aussi appelée norme euclidienne ou de Schur),
et les normes l_p (spécialement pour $p = 1, 2$ et ∞)

$$\| A \|_p = \sup_{x\neq 0} \frac{\| Ax \|_p}{\| x \|_p} \tag{1.2}$$

On peut facilement montrer que pour les normes ci-dessus et n'importe quelles matrices A, B appartenant respectivement à $\mathbf{R}^{n\times m}$ et $\mathbf{R}^{m\times q}$ on a:

$$\| AB \|_p \leq \| A \|_p \| B \|_p \tag{1.3}$$

Remarque:
Pas toutes les normes matricielles satisfont (1.3). Par exemple si on definit la norme de A par

$$\| A \|_\bullet = \max_{i,j} |a_{ij}|$$

et si on prend $A = B = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$, on voit bien que $\| AB \|_\bullet > \| A \|_\bullet \| B \|_\bullet$
Dans ce qui suit, on considérera uniquement les normes qui vérifient (1.3). De la définition de la norme l_p d'une matrice, il s'ensuit que

$$\| A \|_p = \sup_{\|x\|_p=1} \| Ax \|_p \tag{1.4}$$

Du fait de la continuité de la fonction

$$\frac{\| Ax \|_p}{\| x \|_p}$$

sur la sphère unité $\| x \|_p = 1$, il existe toujours un $x^* \in \mathbf{R}^n$ tel que

$$\| Ax \|_p = \| A \|_p \| x^* \|_p \tag{1.5}$$

Les normes F et l_p (spécialement l_1, l_2 et l_∞), satisfont certaines inégalités très utiles dans la suite de ce résumé.

Pour toute matrice $A \in \mathbf{R}^{m \times n}$ on a:

$$\| A \|_2 \leq \| A \|_F \leq \sqrt{n} \| A \|_n \tag{1.6}$$

$$\max_{ij} |a_{ij}| \leq \| A \|_2 \leq \sqrt{nm} \max_{ij} |a_{ij}| \tag{1.7}$$

$$\| A \|_2 \leq \sqrt{\| A \|_1 \| A \|_\infty} \tag{1.8}$$

$$\| A \|_\infty = \max_j \sum_{i=1}^{m} |a_{ij}| \tag{1.9}$$

$$\| A \|_1 = \max_i \sum_{j=1}^{n} |a_{ij}| \tag{1.10}$$

$$\frac{1}{\sqrt{n}} \| A \|_\infty \leq \| A \|_2 \leq \sqrt{m} \| A \|_\infty \tag{1.11}$$

$$\frac{1}{\sqrt{m}} \| A \|_1 \leq \| A \|_2 \leq \sqrt{n} \| A \|_1 \tag{1.12}$$

Une proprieté particulière aux normes F et l_2 est qu'elles sont invariantes par transformations orthogonales, c'est-à-dire pour n'importe quelles matrices orthogonales Q et Z d'ordres appropriés on a:

$$\| QAZ \|_F = \| A \|_F \tag{1.13}$$

et

$$\| QAZ \|_2 = \| A \|_2 \ . \tag{1.14}$$

1.6 Décomposition en valeurs singulières DVS [8, p.5]

Théorème 1.1 *Si A est une matrice réelle d'ordre n, il existe deux matrices orthogonales U et V telles que $U^T AV$ est une matrice diagonale Σ. En plus, on peut toujours choisir U et V de manière que les éléments de Σ vérifient l'inégalité*

suivante:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0,$$

où r est le rang de A.

En particulier si A est régulière on a

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0.$$

Les nombres σ_i sont les valeurs singulières de A. Ce sont les racines carrées positives des valeurs propres (nécessairement positives), de la matrice AA^T qui est symétrique et définie positive.

AA^T est définie positive car pour tout $x \in \mathbf{R}^n$, non nul on a

$$x^T(AA^T)x = (A^Tx, A^Tx) = \| A^Tx \|_2^2 > 0$$

Il découle de ce théorème que A peut s'écrire sous la forme décomposée

$$A = U\Sigma V^T$$

où

$$\Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix},$$

$$U = [u_1, \ldots, u_n]$$

et

$$V = [v_1, \ldots, v_n].$$

Par conséquent

$$Av_i = \sigma_i u_i \tag{1.15}$$

et

$$A^T u_i = \sigma_i v_i \tag{1.16}$$

1.6.1 Interprétation géométrique

Les valeurs singulières σ_i d'une matrice carrée réelle sont précisément les longueurs des branches de l'hyperellipsoïde

$$E = \{y \mid y = Ax, \; \| x \|_2 = 1\}.$$

Par conséquent, il existe une droite L_1 de $\mathbf{R}^n$ qui est étirée (ou rétrécie) du facteur σ_1 quand elle est transformée par A en une autre droite AL_1 de $\mathbf{R}^n$. De même il existe une seconde droite L_n, perpendiculaire à L_1, qui est elle aussi étirée (ou rétrécie) du facteur σ_n quand elle est transformée par A en une autre droite AL_n perpendiculaire à AL_1.

Un cercle de rayon unité pris dans le plan de L_1 et L_n sera transformé par A en une ellipse dont les branches sont de longueurs respectives σ_1 et σ_n. C'est la plus grande déformation qu'un cercle de $\mathbf{R}^n$ puisse subir, quand transformé par A.

De la définition de la norme l_2,

$$\| A \|_2 = \sup_{\|x\|_2 = 1} \| Ax \|_2,$$

et du fait qu'elle est invariante par transformations orthogonales, on a

$$\| A \|_2 = \| U^T \Sigma V \|_2 = \| \Sigma \|_2 = \sigma_1$$

et puisque $A^{-1} = V\Sigma^{-1}U^T$,

$$\| A^{-1} \|_2 = \| V\Sigma^{-1}U^T \|_2 = \| \Sigma^{-1} \|_2 = \sigma_n^{-1}.$$

Comme généralisation du théorème 1.1 on a

Théorème 1.2 *Etant donné une matrice réelle de type* (n, k) *et de rang* r *il existe deux matrices orthogonales* U *et* V *de types respectifs* (n, n) *et* (k, k) *telles que*

$$U^T A V = \Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \\ & 0 & \end{bmatrix}$$

où

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$

1.7 Théorie de la perturbation [25, ch. 4]

Considérons un système linéaire

$$Ax = b. \tag{1.17}$$

Il est admis que la méthode de Gauss est la plus simple et la plus couramment utilisée pour résoudre un tel système, lorsque la matrice A n'a pas de propriété particulière suggérant l'emploi d'une méthode plus élaborée.

S'il apparaît des multiplicateurs trop grands au cours de l'élimination de Gauss il est toujours possible de remédier à cette difficulté en adoptant un choix approprié de pivots. Une autre difficulté pouvant se présenter en pratique est celle d'un système dit "mal conditionné", c'est-à-dire dont la solution est très sensible à de petites perturbations des données initiales A et b .

Considérons d'abord la sensibilité de la solution du système $Ax = b$ à des perturbations du vecteur b. [25, ch. 4].
Si b est remplacé par $b + k$ et $x + h$ est la solution du nouveau système, alors

$$A(x + h) = b + k. \tag{1.18}$$

On obtient par soustraction de (1.17)

$$Ah = k, \tag{1.19}$$

c'est-à-dire $h = A^{-1}k$.
Soit $\| \cdot \|_p$ une norme vectorielle quelconque et $\| \cdot \|_p$ la norme matricielle qui lui est subordonnée; c'est-à-dire

$$\| A \|_p = \sup_{x \neq 0} \frac{\| Ax \|_p}{\| x \|_p}.$$

Nous avons

$$\| h \|_p \leq \| A^{-1} \|_p \| k \|_p . \tag{1.20}$$

Puisque $\| b \|_p \leq \| A^{-1} \|_p \| x \|_p$, on a

$$\frac{\| h \|_p}{\| x \|_p} \leq \| A \|_p \| A^{-1} \|_p \frac{\| k \|_p}{\| b \|_p}. \tag{1.21}$$

Ce qui montre que le nombre $\| A \|_p \| A^{-1} \|_p$ joue le rôle d'une borne supérieure pour la transformation associant à la perturbation relative de la donnée b, la perturbation relative de la solution x. On appelle ce nombre conditionnement de la matrice A (relatif à la norme l_p). Si le conditionnement $\| A \|_p \| A^{-1} \|_p$ est relativement très grand, la majoration (1.21) sera très pessimiste pour la plupart des vecteurs de droite b et des perturbations k. La plupart des b seront tels que

$$\| A^{-1} b \|_p = \| x \|_p \ll \| A^{-1} \|_p \| b \|_p . \tag{1.22}$$

Mais il existe toujours certains choix de b et k pour lesquels la majoration (1.21) est fine, et donc pour lesquels l'effet relatif d'une petite perturbation sur la solution est considérable.

Supposons maintenant que ce soit la matrice A qui soit perturbée par une matrice E:

$$(A + E)(x + h) = b, \tag{1.23}$$

d'où

$$(A + E)h = -Ex. \tag{1.24}$$

Même si A est régulière, ce que nous supposons, $A + E$ peut être singulière si E est quelconque. Écrivons

$$(A + E) = A(I + A^{-1}E).$$

Théorème 1.3 *Si pour une certaine norme matricielle* $\| \cdot \|_p$ *on a* $\| A \|_p < 1$, *alors* $I - A$ *est régulière et son inverse* $(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$ *satisfait*

$$\frac{1}{1 + \| A \|_p} \leq \| (I - A)^{-1} \|_p \leq \frac{1}{1 - \| A \|_p}.$$

On voit bien que $A + E$ est sûrement régulière si

$$\| A^{-1}E \|_p < 1. \tag{1.25}$$

En supposant cette condition satisfaite, on obtient

$$h = -(I + A^{-1}E)^{-1}A^{-1}Ex, \tag{1.26}$$

et donc, d'après le théorème précédent,

$$\| h \|_p \leq \frac{\| A^{-1} E \|_p \| x \|_p}{1 - \| A^{-1} E \|_p} \leq \frac{\| A^{-1} \|_p \| E \|_p \| x \|_p}{1 - \| A^{-1} \|_p \| E \|_p}, \tag{1.27}$$

pouvu que $\| A^{-1} \|_p \| E \|_p < 1$.

Pour la perturbation relative $\| h \|_p / \| x \|_p$, on peut écrire (1.27) sous la forme

$$\frac{\| h \|_p}{\| x \|_p} \leq \frac{\| A \|_p \| A^{-1} \|_p \| E \|_p / \| A \|_p}{1 - \| A \|_p \| A^{-1} \|_p \| E \|_p / \| A \|_p}. \tag{1.28}$$

Ce qui montre, qu'ici encore, le conditionnement de A joue un rôle déterminant dans la majoration de la perturbation relative.

Plus généralement, si l'on considère une perturbation à la fois de A et de b, on aura le système perturbé

$$(A + E)(x + h) = (b + k). \tag{1.29}$$

Si on combine les majorations (1.21) et (1.28) on obtient

Théorème 1.4 *Soit $\|.\|_p$ une norme matricielle. Si $\|E\|_p \|A^{-1}\|_p < 1$ et $\|I\|_p = 1$, alors pour toute norme vectorielle $\|.\|_q$ (compatible avec $\|.\|_p$), on a pour le système (1.21) la majoration suivante.*

$$\frac{\|h\|_q}{\|x\|_q} \leq K(p) \|A\|_p \|A^{-1}\|_p \left(\frac{\|k\|_q}{\|b\|_q} + \frac{\|E\|_p}{\|A\|_p} \right), \tag{1.30}$$

avec

$$K(p) = (1 - \|E\|_p \|A^{-1}\|_p)^{-1}. \tag{1.31}$$

En résumé, puisque $K(p) \geq 1$, la variation relative en x est bornée par $K(p) \|A\|_p \|A^{-1}\|_p$ multiplié par

1. la variation relative de A si b n'est pas perturbé.

2. la variation relative de b si A n'est pas perturbée.

3. la variation relative de A plus la variation relative b si A et b sont tous deux perturbés.

Pour (1.21) on a $K(p) = 1$, alors que pour (1.28) et (1.30), $K(p)$ sera voisin de 1 si $||E||_p$ est suffisamment petite.

Dans tous ces cas on voit bien que la quantité $||A||_p||A^{-1}||_p$, notée $\kappa_p(A)$, est la quantité prépondérante dans l'étude des perturbations de la solution du système linéaire $Ax = b$. Si $\kappa_p(A)$ est petite, l'effet d'une petite perturbation de A ou de b sur la solution x sera modéré ou même négligeable, alors que si $\kappa_p(A)$ est très grand, cet effet peut être considérable pour certains couples (E, k). La majoration (1.21) est la meilleure possible car [8, p. 21], l'égalité peut être atteinte.

Bien entendu, le fait que $\kappa_p(A)$ soit très grand n'entraîne pas que

$$\frac{||h||_p}{||x||_p}$$

soit toujours grand. Mais en général il existera des couples (E, k) pour lesquels, cette perturbation relative sera inadmissible et nécessitera en pratique la prise de précautions efficaces (par exemple calcul en précision double), pour contrôler l'effet des erreurs.

1.8 Conditionnement de matrices

Comme le cas précédent nous l'a montré, la raison qui nous amène à étudier et établir des algorithmes efficaces pour estimer le conditionnement d'une matrice est que dans beaucoup de problèmes matriciels, ce dernier nous donne une idée approximative sur l'effet de la perturbation des données sur la solution exacte.

Le cas le plus populaire en analyse numérique est le système linéaire $Ax = b$; c'est pourquoi il constituera notre unique référence dans ce résumé. On a vu qu'en général quand A et b sont tous les deux perturbés on a

$$\frac{||h||_q}{||x||_q} \leq \frac{||A||_p||A^{-1}||_p}{1 - ||E||_p||A^{-1}||_p}\left(\frac{||k||_q}{||b||_q} + \frac{||E||_p}{||A||_p}\right).$$

Ce résultat nous donne aussi une idée, assez bonne sur l'erreur dans la solution calculée $\hat{x}$, quand le système a été résolu par une méthode stable telle que la méthode de Gauss avec pivot. Car en réalité, ce que la méthode de Gauss nous donne quand

elle est appliquée pour résoudre $Ax = b$, est une solution approchée qui satisfait le système perturbé [12, p.67]:

$$(A + E)\hat{x} = b, \tag{1.32}$$

où E satisfait

$$\|E\|_\infty \leq 8n^3\rho_n\|A\|_\infty u + O(u^2), \tag{1.33}$$

où ρ_n est le facteur de croissance et u l'unité d'arrondi de l'ordinateur utilisé. Une borne rigoureuse pour l'erreur relative sur $\hat{x}$,

$$\frac{\|\hat{x} - x\|_p}{\|x\|_p}$$

peut être obtenue en utilisant (1.30) et (1.33) pourvu que $\kappa_p(A)$ ou bien une borne supérieure de celui-ci soit disponible.

Avant de mieux clarifier cela, rappelons qu'en calcul arithmétique en "virgule flottante", les ordinateurs représentent les nombres sous forme

$$x = \pm 0.d_1 d_2 \ldots d_t \times \beta^e,$$

où les d_i sont des entiers en base β (par exemple $\beta = 2$), et où le nombre t, fixé une bonne fois pour toutes, par exemple par les caractéristiques de l'ordinateur, est le nombre de chiffres en base β qui représentent la mantisse de x. Par exemple les ordinateurs pour lesquels $t = 15$, représentent le nombre $x = 732.67$ en base $\beta = 2$, comme

$$x = 0.101101110010101 \times 2^{10},$$

et l'erreur commise sur x est inférieure à $2^{-t} = 2^{-15}$. Par conséquent, la solution approchée (1.32) donnée par la méthode de Gauss satisfait: voir [2]

$$\frac{\|\hat{x} - x\|_p}{\|x\|_p} \sim \beta^{-t}\kappa_p(A), \tag{1.34}$$

où

$$\frac{\|E\|_p}{\|A\|_p} \sim \beta^{-t}. \tag{1.35}$$

Maintenant supposons que $\kappa_p(A) \sim \beta^q$. Donc

$$\frac{\|\hat{x} - x\|_p}{\|x\|_p} \sim \beta^{q-t}. \tag{1.36}$$

Heuristique [12, p. 72]

La méthode de Gauss produit une solution approchée $\hat{x}$, avec environ $t \log_{10} \beta - \log_{10}[\kappa_\infty(A)]$ chiffres significatifs exacts. [12, p. 71]
Comme on a déjà vu dans la section (1.5), $\kappa_2(A)$, est donnée par $\kappa_2(A) = \sigma_1/\sigma_n$ où σ_1 est la plus grande valeur singulière de A et σ_n est la plus petite valeur singulière de A car $\|A^{-1}\|_2 = \sigma_n^{-1}$ et $\|A\|_2 = \sigma_1$. L'estimation du conditionnement d'une matrice, est requise dans beaucoup d'autres domaines de l'analyse numérique matricielle. Quelques exemples importants sont:
l'optimisation [4, p. 55], [9], [10, pp. 135-320], le calcul des moindres carrées [12, chap. 6], [19], [20], l'estimation du conditionnement des valeurs et vecteurs propres [24], le calcul des racines carrées d'une matrice [17] et de l'exponentielle d'une matrice [21], les solutions des équations matricielles de Sylvester et de Lyapunov [1], [11], [15], le calcul avec les matrices creuses [7], [13] et la solution numérique d'équations differentielles et intégrales.

Dans ces domaines d'application, la matrice en question est presque toujours ou bien déjà triangulaire, ou bien décomposée en une expression qui contient un facteur triangulaire. Dans les deux cas le conditionnement de A peut être obtenu à partir de celui du facteur triangulaire. C'est pourquoi d'ailleurs, dans les chapitres qui suivront, l'étude du conditionnement d'une matrice sera restreinte au cas d'une matrice triangulaire.

Chapitre 2

Techniques pour estimer le conditionnement d'une matrice.

2.1 Introduction

Comme il a déjà été souligné dans le chapitre précédent, après la résolution d'un système linéaire d'équations algébriques $Ax = b$, ce qui est généralement fait par la méthode gaussienne qui produit une solution approchée $\hat{x}$, il est toujours important d'avoir une idée sur l'ordre de grandeur de $\kappa_p(A)$. La connaissance de cette quantité nous permet d'avoir une idée approximative sur la précision relative de $\hat{x}$.

Différents algorithmes et techniques pour estimer $\kappa_p(A)$, ainsi que les résultats numériques concernant leur performance en pratique seront le sujet de ce deuxième chapitre.

2.2 Généralités

Parmi beaucoup d'autres critères, la préférence d'une méthode, ou plus exactement de l'algorithme qui en découle, est basée essentiellement sur sa fiabilité et le nombre d'opérations algébriques de la forme $s = s + a_{ik}x_k$, qu'il utilise.

Dans la littérature anglo-saxone, une telle expression est appelée "flop", ce qui désormais, remplacera le terme opération algébrique.

Pour plus de details sur la définition du mot flop, voir [12, p. 32].

Puisqu'en pratique, les normes les plus utilisées sont les normes l_1, l_2 et l_∞, notre étude se restreindra à ces dernières.

Avant d'exposer les différentes méthodes connues, on rappelle que pour toute matrice carrée

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \in \mathbf{R}^{n \times n},$$

on a

$$||A||_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

$$||A||_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

$$||A||_2 = \rho(A^T A)^{1/2},$$

où ρ est le rayon spectral de $A^T A$.

Le conditionnement de A par rapport à la norme p est

$$\kappa_p(A) = ||A||_p \, ||A^{-1}||_p \quad p = 1, 2 \text{ et } \infty.$$

Pour $p = 2$, on a déjà vu que $\kappa_2(A) = \sigma_1/\sigma_n$, où σ_1 et σ_n sont respectivement la plus grande et la plus petite valeur singulière de A.

Maintenant si $p = 1$ ou $p = \infty$, on voit bien que le calcul de $||A||_p$ ne pose aucune difficulté, tandis que celui de $||A||_2$ peut toujours être estimé ou bien par l'inégalité (1.12), ou bien par la méthode des puissances [25, p. 570].

Dans les trois cas, la difficulté provient du facteur $||A^{-1}||_p$, $p = 1, 2, \infty$. Car, non seulement la connaissance de A^{-1} n'est pas toujours nécessaire, mais le coût de son calcul est de l'ordre de $n^3/3 + O(n^2)$ flops; ce qui est assez cher comparativement à la décomposition de Gauss.

Penser pouvoir estimer le conditionnement spectral $\kappa_2(A)$ par $\hat{\sigma}_1/\hat{\sigma}_n$, où $\hat{\sigma}_1$ et $\hat{\sigma}_n$ sont calculées par le sous-programme [SVD] de, LINPACK [6] ou EISPACK [23], ne résoudra pas le problème non plus. Car ce dernier utilise approximativement 15 fois plus de flops que l'élimination gaussienne.

Ce qui nous intéresserait plutôt, ce serait des méthodes qui utiliseraient un ordre de $O(n^2)$ flops au plus.

Finalement, puisque A est généralement supposée avoir été décomposée en une expression contenant un facteur triangulaire, que son conditionnement peut être obtenu à partir de celui du facteur triangulaire et que la difficulté vient du terme $||A^{-1}||_p$, $p = 1, 2, \infty$, il est clair qu'il suffira de s'intéresser au facteur $||T^{-1}||_p$, $p = 1, 2, \infty$, où T est une matrice triangulaire.

Forsythe et Moler [8] ont proposé une méthode intéressante pour estimer $\kappa_\infty(A)$. Elle est basée sur les itérations effectuées sur la solution approchée $\hat{x}$ pour améliorer sa précision.

2.3 Estimation de $\kappa_\infty(A)$ par la méthode itérative [8]

Soit le système linéaire

$$Ax = b.$$

En supposant que la solution approchée $\hat{x}$ soit calculée au moyen de la décomposition gaussienne $PA = LU$, ces deux auteurs ont posé

$$\kappa_\infty(A) \simeq \beta^t ||\hat{z}||_\infty / ||\hat{x}||_\infty,$$

où $\hat{z}$ est calculé en résolvant les deux systèmes $Lw = Pr$ et $Uz = w$ avec le résidu $r = b - A\hat{x}$ calculé en double précision.

Ici, β représente la base en virgule flottante utilisée par l'ordinateur et t, le nombre de chiffres significatifs avec lequel sont repésentés les nombres.

Malgré l'efficacité de cette méthode, son utilisation est limitée à cause des problèmes de portabilité associés au calcul de r en double précision et la nécessité d'avoir n^2 cases mémoires additionnelles.

Sa fiabilité n'a pas été prouvée mais il apparaît qu'elle fonctionne assez bien.

Le coût de cette méthode est de l'ordre de $O(n^2)$ flops.

Plus tard, une méthode intéressante pour estimer $||A^{-1}||_p$, $p = 1, 2, \infty$, donc $\kappa_p(A)$, a été proposée par A.K. Cline, C.B. Moler, G.W. Stewart et J.H. Wilkinson, [3].

2.4 Méthode de Cline, Moler, Stewart et Wilkinson

Si y est la solution approchée du système linéaire $Ax = b$, c'est-à-dire $(A+E)y = b$, où E est une petite perturbation de A, alors

$$\frac{||y-x||}{||y||} \leq \epsilon ||A||_p ||A^{-1}||_p, \quad \text{où} \quad \epsilon = \frac{||E||}{||A||}. \tag{2.1}$$

Si c'est le vecteur des données b qui est perturbé, c'est-à-dire $Ay = b + e$, on a

$$\frac{||y-x||}{||x||} \leq \epsilon \kappa_p(A), \quad \text{où} \quad \epsilon \geq \frac{||e||}{||b||}. \tag{2.2}$$

L'analyse en termes de décomposition en valeurs singulières de A donne:
Soit

$$A = U\Sigma V^T, \tag{2.3}$$

où U et V sont orthogonales et $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$, où $\sigma_1 \geq \cdots \geq \sigma_n > 0$ sont les valeurs singulières de A.

De (2.3) on a

$$Av_i = \sigma_i u_i \tag{2.4}$$

et

$$A^T u_i = \sigma_i v_i. \tag{2.5}$$

Exprimons les vecteurs b et e en fonction des vecteurs orthogonaux u_i de la manière suivante:

$$b = ||b|| \sum_{i=1}^{n} \alpha_i u_i, \quad \text{où} \quad \sum_{i=1}^{n} \alpha_i^2 = 1, \tag{2.6}$$

et

$$e = \epsilon ||b|| \sum_{i=1}^{n} \beta_i u_i, \quad \text{où} \quad \sum_{i=1}^{n} \beta_i^2 = 1. \tag{2.7}$$

De (2.4) et (2.5) on a

$$x = ||b|| \sum_{i=1}^{n} (\alpha_i/\sigma_i) v_i, \tag{2.8}$$

et

$$y - x = \epsilon ||b|| \sum_{i=1}^{n} (\beta_i/\sigma_i) v_i. \tag{2.9}$$

D'où

$$\frac{||y-x||}{||x||} = \epsilon\sigma_1/\sigma_n = \epsilon\kappa_2(A), \tag{2.10}$$

uniquement dans le cas où $b = ||b||u_1$ et $e = \epsilon||b||u_n$.

Quand σ_1/σ_n est grand, $||y-x||/\epsilon||x||$ est de l'ordre de l'unité pour n'importe quel e inférieur ou égal à $\epsilon||b||$, à l'exception du cas où α_n est très petit. Puisque aussi

$$||x||/||b|| = \left(\sum_{i=1}^{n}(\alpha_i/\sigma_i)^2\right)^{1/2}, \tag{2.11}$$

ce rapport sera de l'ordre de σ_n^{-1} à moins que α_n ne soit exceptionnellement très petit.

Maintenant on voit bien que pour $Rx = b$, où R est triangulaire, on a

$$\max_{b\neq 0}\left(\frac{||x||_2}{||b||_2}\right) = \sigma_n^{-1} = ||R^{-1}||_2. \tag{2.12}$$

Cette borne est bien sûr atteinte quand $b = u_n$.

La décomposition en valeurs singulières nous suggère de résoudre les deux systèmes

$$R^T x = b \qquad \text{et} \qquad Ry = x. \tag{2.13}$$

Si on pose $b = \sum_{i=1}^{n}\alpha_i v_i$, on aura

$$x = \sum_{i=1}^{n}(\alpha_i/\sigma_i)u_i \tag{2.14}$$

et

$$y = \sum_{i=1}^{n}(\alpha_i/\sigma_i^2)v_i. \tag{2.15}$$

Pourvu que la composante α_n de b ne soit pas trop petite, le vecteur y sera dans la plupart des cas dominé par la composante de v_n.

On a

$$\frac{||y||_2}{||x||_2} = \left[\sum_{i=1}^{n}(\alpha_i/\sigma_i^2)^2 / \sum_{i=1}^{n}(\alpha_i/\sigma_i)^2\right]^{1/2}. \tag{2.16}$$

Si α_n/σ_n n'est pas trop petit relativement aux autres facteurs, le rapport donnera une assez bonne approximation de σ_n^{-1}. On voit aussi que cette estimation

est d'autant meilleure que $\sigma_n \ll \sigma_1$, ce qui est bien sûr le cas quand A est mal conditionnée.

Si on prend un b arbitraire, la probabilité d'obtenir une bonne estimation de σ_n^{-1} est grande. Afin d'augmenter cette probabilité naturelle, ces quatres auteurs ont proposé de faire un choix judicieux de b.

Ce dernier se fait de manière que la solution x du système $R^T x = b$ donne un rapport $||x||/||b||$ aussi grand que possible.

L'idée suivante se présente d'elle-même.

Choisir $b_s = \pm 1$, où les signes des b_s sont determinés à l'étape où x_s est calculé par la relation

$$x_s r_{ss} = b_s - (r_{1s}x_1 + \cdots r_{s-1,s}x_{s-1}). \tag{2.17}$$

Ce signe est choisi comme l'opposé de celui du produit scalaire entre parenthèses.

Cette méthode serait bonne si ce n'était le contre-exemple suivant:

Soient $n = 4$ et

$$R^T = \begin{bmatrix} I & 0 \\ kE & I \end{bmatrix} \text{ où } E = \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}.$$

Donc

$$R^{-T} = \begin{bmatrix} I & 0 \\ -kE & I \end{bmatrix} \text{ et } ||R^T||_1 = ||R^{-T}||_1 = 1 + 2k.$$

Si k est grand, alors $||R^T||_1$ et $||R^{-T}||_1$ sont grands. Si on pose $b_1 = +1$, ce qui donne $x_1 = 1$, les deux signes possibles de b_2 donneront $|x_2| = 1$.

Si on pose $b_2 = +1$, alors $|x_3| = |x_4| = +1$ et on n'aura pas une grande norme pour x. Si on prend $b_2 = -1$, on aura la même situation avec

$$E = \begin{bmatrix} +1 & +1 \\ +1 & +1 \end{bmatrix}.$$

Pour le premier E, si on prend $b_s = +1$ à chaque étape on aura

$$R^T b = b \qquad \text{et} \qquad Rb = b.$$

Ceci n'indique pas si R est bien ou mal conditionnée.

Il est clair que la faiblesse de cette méthode réside dans le fait que la décision sur le choix du signe de b_s est basée sur un critère local.

Les mêmes auteurs ont remédié à cela en ajoutant à la même méthode, une technique qui consiste à regarder en avant, [look-ahead], avant de décider du signe de b_s.

Cela conduit à la stratégie suivante:

Chaque x_i $(i > s)$ est déterminé par la relation

$$r_{ii}x_i = (-r_{1i}x_1 - \cdots - r_{s-1,i}x_{s-1}) + (-r_{si}x_s - \cdots - r_{i-1,i}x_{i-1} + b_i) \quad (2.18)$$

où l'expression de droite est divisée en deux parties. La première partie est calculée une fois que $x_1, \ldots, x_{s-1}$ sont calculées. On note cette partie par $p_i^{(s-1)}$. Juste avant de donner un signe à b_s et calculer x_s, supposons qu'on a déjà calculé et emmagasiner les quantités $p_i^{(s-1)}$ $(i = s, \ldots, n)$. Maintenant les deux valeurs possibles de x_s sont données par

$$r_{ss}x_s = -p_s^{(s-1)} + b_s = -p_s^{(s-1)} \pm 1. \quad (2.19)$$

On notera ces deux valeurs possibles de x_s par x_s^+ et x_s^-. Une fois calculées, ces deux valeurs de x_s nous serviront pour renouveler les quantités $p_i^{(s-1)}$ par les quantités $p_i^{(s)}$ qui seront déterminées de la manière suivante.

On écrit

$$p_i^{(s)+} = p_i^{(s-1)} + r_{si}x_s^+ \qquad \text{et} \qquad p_i^{(s)-} = p_i^{(s-1)} + r_{si}x_s^-. \quad (2.20)$$

Notre décision sur le signe de b_s dépendra des grandeurs de ces $|p_i^{(s)}|$ et de $|x^s|$ lui-même.

Comme critère on peut prendre $b_s = \pm 1$ selon que

$$|-p_s^{(s-1)}+1|+\sum_{i=s+1}^{n}|p_i^{(s)+}| \quad \geq \quad \text{ou} \quad < \quad |-p_s^{(s-1)}-1|+\sum_{i=s+1}^{n}|p_i^{(s)-}|. \quad (2.21)$$

L'algorithme qui découle de cette méthode est le suivant:

Algorithme 1

1^e étape: Choisir b de manière que la norme de la solution de $R^T x = b$ soit grande relativement à celle de b.

2^e étape: Résoudre $Ry = x$.

3^e étape: Poser $\kappa_1(A) = ||R||_1||y||_1/||x||_1$.

Pour la motivation de la deuxième étape, l'idée est que si

$$||x||/||b|| \ (\approx ||R^{-T}||)$$

est grand, alors

$$||y||/||x|| \ (\approx ||R^{-1}||)$$

est au moins aussi grand.

Après beaucoup d'expérimentations, O'Leary a suggéré de modifier cet algorithme en prenant

$$||R^{-T}||_1$$

comme

$$\max\{||x||_\infty/||b||_\infty, ||y||_1/||x||_1\},$$

puisque

$$||R^{-1}||_1 = ||R^{-T}||_\infty \simeq ||x||_\infty/||b||_\infty.$$

Le coût de cet algorithme est de $(5/2)n^2$ flops. Pour en tester la performance, O'Leary [22] a calculé la moyenne sur les rapports

$$r = \frac{||A^{-1}||_1 \text{estimé}}{||A^{-1}||_1 \text{exact}},$$

pour 100 matrices de dimensions variant de 5 à 50, dont les éléments sont choisis d'une distribution uniforme entre -1 et +1.

n	r moyen
5	0.69
10	0.60
20	0.52
40	0.43

Table 2.1: Moyenne des r

Les auteurs ont considéré que cette méthode est fiable. Si le conditionnement de A est grand, son estimation est grande.

Ce qui est toutefois décevant, c'est que, si on prend $x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ et que l'on résout $Ay = x$, on aura les résultats ci-dessous.

Posant

$$s = \frac{\sum_{i=1}^{n} |a_{i1}^{-1}|}{||A^{-1}||_1},$$

où $||A^{-1}||_1$ est approximée par la somme des valeurs absolues des éléments de la première colonne de A^{-1}, $||y||_1$, on a les résultats suivants: [W. W. Hager]

n	r moyen
5	0.61
10	0.55
20	0.42
40	0.40

Table 2.2: Moyenne des r

On remarque bien que cette dernière méthode est presque aussi performante que la première.

Un peu plus tard, Cline, Conn et Van Loan [2], ont généralisé cette méthode en lui incorporant une technique, qui, dans la détermination des composantes b_s du vecteur b de la première étape de l'algorithme 1, nous permet de regarder en arrière, [look-behind], et de changer les composantes b_s déjà choisies.

Pour un système linéaire

$$Ty = d,$$

cette méthode nous permet de faire un choix du vecteur d, d'une meilleure manière que dans l'algorithme précédent, afin que la solution y du système ci-dessus soit grande en norme.

Pour illustrer cette méthode, qu'on appellera "méthode avec regard rétrograde", supposons que la matrice T soit d'ordre 6 et que les trois premières composantes d_1, d_2 et d_3 soient connues et vérifient

$$d_1^2 + d_2^2 + d_3^2 = 1.$$

De plus, supposons qu'on a résolu le système

$$\begin{bmatrix} t_{11} & 0 & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}, \tag{2.22}$$

et qu'on a calculé les quantités

$$p_4 = t_{41} y_1 + t_{42} y_2 + t_{43} y_3,$$

$$p_5 = t_{51} y_1 + t_{52} y_2 + t_{53} y_3,$$

$$p_6 = t_{61} y_1 + t_{62} y_2 + t_{63} y_3.$$

On veut maintenant déterminer les deux paramètres

$$c = \cos a$$

et

$$s = \sin a,$$

de manière que si

$$\begin{bmatrix} t_{11} & 0 & 0 & 0 \\ t_{21} & t_{22} & 0 & 0 \\ t_{31} & t_{32} & t_{33} & 0 \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} y_1' \\ y_2' \\ y_3' \\ y_4' \end{bmatrix} = \begin{bmatrix} sd_1 \\ sd_2 \\ sd_3 \\ c \end{bmatrix}, \tag{2.23}$$

l'expression

$$\sum_{i=1}^{4} (y_i')^2 + \sum_{i=5}^{6} (p_i')^2, \tag{2.24}$$

où $p_i' = sp_i + t_{i4} y_4'$, $(i = 5, 6)$ soit maximisée.

Les quantités p_i' représentent des renouvellements des quantités p_i.

Remarquons que le vecteur de droite du système (2.23) reste de longueur unité et que la solution de ce système est donnée par

$$y_4' = (c - sp_4)/t_{44}$$

et

$$y_i' = sy_i \quad (i = 1, 2, 3).$$

En général, à la k-ème étape quand la composante b_k est calculée, on regarde en arrière pour réviser les composantes $d_1, \cdots, d_{k-1}$, déjà calculées, et on regarde en avant pour anticiper l'effet sur les quantités $p_{k+1}, \cdots, p_n$. L'expression (2.24) est une fonction de a qu'on notera par $\Phi(a)$. Le paramètre a qui maximise $\Phi(a)$ peut être facilement déterminé en dérivant la fonction Φ par rapport à a.

Finalement, une fois qu'on obtient la solution du système (2.23), on a une approximation de la plus petite valeur singulière de T.

$$\hat{\sigma}_n = ||y||_2^{-1} \simeq \sigma_n.$$

Si au lieu de maximiser la fonction Φ, on la minimise, la quantité $||y||_2^{-1}$, donnera une approximation de la plus grande valeur singulière de T.

$$\hat{\sigma}_1 = ||y||_2^{-1} \simeq \sigma_1.$$

L'application de cette technique dans le cas de la norme l_1 se fait comme suit:

Reprenons la matrice d'ordre 6, utilisée pour illustrer cette technique dans le cas de la norme l_2.

Supposons que les composantes d_1, d_2 et d_3 soient connues et vérifient

$$|d_1| + |d_2| + |d_3| = 1.$$

Supposons aussi que le système (2.23) soit résolu et que les quantités p_4, p_5 et p_6 soient calculées. Cette fois-ci on cherche un paramètre $\lambda \in [0, 1]$ de manière que si

$$\begin{bmatrix} t_{11} & 0 & 0 & 0 \\ t_{21} & t_{22} & 0 & 0 \\ t_{31} & t_{32} & t_{33} & 0 \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} y_1' \\ y_2' \\ y_3' \\ y_4' \end{bmatrix} = \begin{bmatrix} \lambda d_1 \\ \lambda d_2 \\ \lambda d_3 \\ 1 - \lambda \end{bmatrix}, \tag{2.25}$$

l'expression

$$\sum_{i=1}^{4} |y_i'| + \sum_{i=5}^{6} |p_i'|, \tag{2.26}$$

où $p_i' = \lambda p_i + t_{i4}\, y_4'$, $(i = 5, 6)$, soit maximisée. Puisque l'expression (2.26) est une fonction convexe de λ, il suffit de vérifier ses valeurs en $\lambda = 0$ et $\lambda = 1$.

Dans ce cas, la solution y du système $Ty = d$, nous donnera une estimation de $||T^{-1}||_1$:

$$||y||_1 \simeq ||T^{-1}||_1.$$

Les mêmes auteurs ont incorporé à cette méthode une technique qui consiste à diviser pour conquérir, [divide and conquer]. Pour l'illustrer considérons deux matrices triangulaires inférieures, $T_{11} \in \mathbf{R}^{p\times p}$ et $T_{22} \in \mathbf{R}^{q\times q}$.

Supposons que les deux systèmes

$$T_{11}y_1 = d_1 \qquad \text{où} \qquad \|d_1\|_2 = 1$$

et

$$T_{22}y_2 = d_2 \qquad \text{où} \qquad \|d_2\|_2 = 1,$$

soient résolus.

On cherchera $c = \cos a$ et $s = \sin a$, de manière que si

$$\begin{bmatrix} T_{11} & 0 \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} cd_1 \\ sd_2 \end{bmatrix},$$

l'expression

$$\Phi(a) = \|z_1\|_2^2 + \|z_2\|_2^2,$$

soit maximisée.

Définissons $\omega \in \mathbf{R}^q$ par

$$T_{22}\omega = T_{21}y_1.$$

La solution du système précédent est donnée par

$$z_1 = cy_1$$

et

$$z_2 = sy_2 - c\omega.$$

Par conséquent

$$\Phi(a) = c^2[\|y_1\|_2^2 + \|\omega\|_2^2] - 2scy_2^T\omega + s^2\|y_2\|_2^2.$$

Le paramètre a qui maximise $\Phi(a)$ est obtenu en dérivant cette dernière. Pour les détails sur les algorithmes dérivés de ces méthodes voir [2].

Notons E1 l'algorithme "diviser pour conquérir" et E2 l'algorithme avec regard rétrograde.

E1 et E2 ont été testés dans le cas de la norme l_2. Ces tests ont été effectués sur 1000 matrices triangulaires dont les ordres varient de 5 à 50, avec 100 de chaque ordre. Les éléments de celles-ci ont été pris aléatoirement dans l'intervalle [-1, +1]. Soit $q_n = \hat{\sigma}_n^{-1}/\sigma_n^{-1}$.

Algorithme		E1	E2
>	<=	q_n	q_n
.9	1.0	65.1%	62.6%
.8	.9	12.4%	11.6%
.6	.7	4.7%	3.5%
.5	.6	4%	4.4%
.0	.5	7.6%	10.6%

Table 2.3: Pourcentages des réussites

Avec la méthode E1, 65.1% des rapports q_n sont compris entre 0.9 et 1.

On remarque que les résultats obtenus avec l'algorithme E1 sont légèrement meilleurs que ceux obtenus avec E2.

Pour les matrices générées par la décomposition QR d'une matrice pleine A, les résultats obtenus sont de loin meilleurs que ceux présentés ci-dessus. Pour plus de détails, voir [2].

Plus tard, William W. Hager [14], a présenté une approche tout à fait différente pour estimer $||A^{-1}||_1$ avec une grande précision. Cette dernière méthode est basée sur l'optimisation convexe.

2.5 Méthode par l'optimisation convexe [14]

Pour une matrice B de $\mathbf{R}^{n\times n}$ la norme l_1 de B est la valeur maximale de la fonction convexe

$$f(x) = \|Bx\|_1 = \sum_{i=1}^{n} |\sum_{j=1}^{n} b_{ij}x_j|, \tag{2.27}$$

sur l'ensemble convexe

$$S = \{x \in \mathbf{R}^n \mid \|x\|_1 \leq 1\}. \tag{2.28}$$

Des résultats sur la convexité, il s'ensuit que f atteint son maximum en l'un des sommets e_j, $j = 1, \ldots, n$, de S, où e_j désigne la j-ème colonne de la matrice identité.

Les $2n$ sommets de S sont

$$\{\pm e_j \; j = 1, \cdots, n\}.$$

Puisque f est convexe, elle satisfait l'inégalité

$$f(y) \geq f(x) + \partial f(x)(y - x) \text{ pour tout } x, y \in \mathbf{R}^n. \tag{*}$$

On note $\partial f(x)$ le sous-gradient de f en x.

Si

$$\sum_{j=1}^{n} b_{ij}x_j \neq 0 \quad \text{pour tout } i,$$

alors $\partial f(x)$ est le vecteur gradient usuel. Si pour tout $i = 1, \cdots, n$, on définit

$$\xi_i = \begin{cases} +1 & \text{si } \sum_{i=1}^{n} b_{ij}x_j \geq 0 \\ -1 & \text{autrement,} \end{cases}$$

on aura

$$\partial f(x) = \xi^T B.$$

Dans le cas où $B = A^{-1}$, calculer $\partial f(x)$ revient à résoudre les deux systèmes:

$$Ay = x, \; A^T z = \xi,$$

où

$$\xi_i = \begin{cases} +1 & \text{si } y_i \geq 0 \\ -1 & \text{autrement,} \end{cases}$$

et $\partial f(x) = z^T$.

L'algorithme pour estimer $||B||_1$ commence en un point x de la frontière de S.

On trouvera alors un indice j pour lequel $|\partial f(x)_j| = \max_i |\partial f(x)_i|$.

Si $|\partial f(x)_j| \leq \partial f(x)x$, arrêter.

Si $|\partial f(x)_j| > \partial f(x)x$, par l'inégalité (*) et le fait que $f(e_j) = f(-e_j)$, on conclut que $f(e_j) > f(x)$.

Il suffit alors de remplacer x par e_j et de répéter le processus.

L'algorithme qui découle de cette méthode est:

Algorithme 2

Choisir x tel que $||x||_1 = 1$ (e.g. $x = n^{-1}(1, \ldots, 1)^T$).
Répéter
  Résoudre $Ay = x$.
  Former ξ où $\xi_i = \begin{cases} +1 & \text{si} \quad y_i \geq 0, \\ -1 & \text{si} \quad y_i < 0. \end{cases}$
  Résoudre $A^T z = \xi$. (*)
  Si $||z||_\infty \leq z^T x$ donc
    $\gamma = ||y||_1 \quad (= f(x))$
    arrêter
  $x = e_j \quad$ où $\quad |z_j| = ||z||_\infty$.

L'algorithme peut être interprété de la manière suivante.

On peut montrer que le vecteur z calculé à l'étape (*) est un sous-gradient de la fonction f en x.

Il découle des propriétés de la convexité que $f(\pm e_j) \geq f(x) + z^T(\pm e_j - x)$, $j = 1, \ldots, n$, de manière que si $|z_j| > z^T x$, pour un certain j, alors f peut être augmentée en se déplacant du point x au sommet e_j de S.

Par contre si $||z||_\infty \leq z^T x$ et si $y_j \neq 0 \ \forall j$, on peut prouver que x est un maximum local pour f sur S.

La condition $y_j \neq 0$ pour tout j, assure que l'ensemble des sous-gradients de f en x est un singleton dont l'élément unique est le vecteur gradient usuel.

Hager a construit un contre-exemple pour cet algorithme; ce qui n'est pas d'une grande importance car la précision de l'estimation est dans l'ensemble meilleure que toutes les précédentes.

Le coût de cet algorithme est de $2n^2$ ou $3n^2$ flops.

Pour tester la performance de celui-ci, l'auteur a calculé le rapport

$$t_1 = \frac{\|A^{-1}\|_1 \text{estimé}}{\|A^{-1}\|_1 \text{exact}}, \tag{2.29}$$

pour 200 matrices de même dimension, dont les éléments sont pris d'une distribution uniforme sur l'intervalle [-1,1].

Le choix de x initial a été $n^{-1}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$. On voit que la fiabilité de cet algorithme

n	t_1 moyen
5	0.96
10	0.97
20	0.98
40	0.97
80	0.98

Table 2.4: Moyenne des t_1

est indépendante de la dimension n de la matrice. La moyenne du nombre de cycles nécessaires pour atteindre le maximum local est de 2.1.

Une amélioration pour augmenter la précision de l'estimation de $||A^{-1}||_1$ a été donnée par l'auteur. Elle consiste à appliquer la méthode précédente pour certains sous-espaces appropriés. Pour une description detaillée voir [14].

Dans le chapitre 3 qui suit, on présente une autre technique, pour estimer le conditionnement spectral de A, ainsi qu'une comparaison des différentes méthodes exposées jusqu'alors.

Chapitre 3

Estimation du conditionnement spectral d'une matrice

3.1 Introduction

Puisque dans beaucoup de problèmes linéaires, la norme vectorielle euclidienne l_2 et la norme matricielle induite sont utilisées, ce chapitre a pour but de présenter une technique économique et facilement incorporable pour estimer le conditionnement spectral d'une matrice, et de comparer sa performance en pratique à celles des méthodes présentées au chapitre précédent.

3.2 Généralités
3.3 Estimation de $||R^{-1}||_2$ quand R est triangulaire
3.4 Estimation de $||A^{-1}||_2$ quand A est pleine
3.5 Comparaison des différentes méthodes et conclusion

3.2 Généralités

Avant de résoudre un système linéaire $Ax = b$, il est préférable de lui faire subir une certaine mise à l'échelle ("scaling"), destinée à réduire si possible le conditionnement de la matrice du système.

Wilkinson (1965) recommande d'"équilibrer" la matrice A d'après un processus que nous allons définir et qui s'accompagne presque toujours d'une diminution du conditionnement de la matrice du système.

Définition 3.1 *On dit qu'une matrice A est équilibrée si toutes ses lignes et ses colonnes ont une longueur de l'ordre de l'unité, et si l'on a de plus*

$$\frac{1}{\beta} \leq \max_{1 \leq j \leq n} |a_{ij}| \leq 1 \qquad (i = 1, \ldots, n)$$

et

$$\frac{1}{\beta} \leq max_{1 \leq i \leq n} |a_{ij}| \leq 1 \qquad (j = 1, \ldots, n),$$

où β est la base du système d'arithmétique en point flottant utilisé par l'ordinateur.

Pour résoudre un système linéaire, la décomposition gaussienne avec pivotage $PA = LU$ est en général la plus utilisée, à moins que la matrice A ne posséde des propriétés particulières.

Par exemple, si A est une matrice bande symétrique et définie positive, il est préférable d'utiliser la décomposition de Cholesky, $A = LL^t$, où L est triangulaire inférieure.

Dans ce dernier cas on doit souligner que $\kappa_2(A) = \kappa_2(L)^2$. Ceci ramène l'estimation du conditionnement spectral de A à celui de L . De même si on a la décomposition QR de A on aura $\kappa_2(A) = \kappa_2(R)$.

On a aussi vu au chapitre 2 que dans tous les cas, la dificulté pour estimer $\kappa_p(A)$ provient du terme $||A^{-1}||_p$. Le facteur $||A||_2$ peut toujours être estimé par l'une des inégalités (1.6), (1.11) ou (1.12).

Par conséquent, on se contentera d'estimer $||R^{-1}||_2$, où R est une matrice triangulaire .

3.3 Estimation de $||R^{-1}||_2$ quand R est triangulaire

Reconsidérons encore une fois l'analyse du problème en termes de la décomposition en valeurs singulières:

Soit

$$R = U\Sigma V^t, \tag{3.1}$$

où

$$U = [u_1, \ldots, u_n], V = [v_1, \ldots, v_n]$$

sont orthogonales, et

$$\Sigma = \text{diag}\,(\sigma_1, \ldots, \sigma_n), \qquad \text{avec } \sigma_1 \geq \ldots \geq \sigma_n.$$

Soit le système

$$Rx = b, \tag{3.2}$$

où R est une matrice triangulaire inversible.

Par définition de la norme matricielle on a

$$||R^{-1}||_2 = \sup_{b \neq 0} \frac{||x||_2}{||b||_2}. \tag{3.3}$$

On a vu au chapitre précédent que l'algorithme avec regard rétrograde nous permet de faire un choix du vecteur b de manière que la solution x du système (3.2) soit grande en norme.

Cette méthode peut être améliorée au moyen d'un regard doublement rétrograde, de la manière suivante:

Supposons ici encore que $n = 6$ et que b_1, b_2 et b_3 soient connus et vérifient la relation

$$b_1^2 + b_2^2 + b_3^2 = 1.$$

Supposons aussi qu'on ait déjà résolu le système

$$\begin{bmatrix} r_{11} & 0 & 0 \\ r_{21} & r_{22} & 0 \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

et calculé les quantités

$$\begin{aligned} p_4 &= r_{41}x_1 + r_{42}x_2 + r_{43}x_3 \\ p_5 &= r_{51}x_1 + r_{52}x_2 + r_{53}x_3 \\ p_6 &= r_{61}x_1 + r_{62}x_2 + r_{63}x_3 \end{aligned}$$

Maintenant, on cherchera deux paramètres α_1 et α_2 de manière que si

$$\begin{bmatrix} r_{11} & 0 & 0 & 0 & 0 \\ r_{21} & r_{22} & 0 & 0 & 0 \\ r_{31} & r_{32} & r_{33} & 0 & 0 \\ r_{41} & r_{42} & r_{43} & r_{44} & 0 \\ r_{51} & r_{52} & r_{53} & r_{54} & r_{55} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ x'_5 \end{bmatrix} = \begin{bmatrix} b_1 \sin\alpha_1 \sin\alpha_2 \\ b_2 \sin\alpha_1 \sin\alpha_2 \\ b_3 \sin\alpha_1 \sin\alpha_2 \\ \cos\alpha_1 \sin\alpha_2 \\ \cos\alpha_2 \end{bmatrix},$$

l'expression

$$\Phi(\alpha_1, \alpha_2) = \sum_{i=1}^{5} (x'_i)^2 + (p'_6)^2 \quad \text{où} \quad p'_6 = \sum_{i=1}^{5} r_{6i}x'_i, \tag{*}$$

sera maximisée.

Cela consiste bien sûr à déterminer les composantes b_i et les solutions x_i, deux à deux. Si à la fin du processus il ne reste que la seule composante b_n à déterminer, ce qui est le cas quand n est pair, on devra appliquer l'algorithme avec regard rétrograde.

L'expression (*) est une fonction des paramètres α_1 et α_2. Notons cette dernière par $\Phi(\alpha_1, \alpha_2)$.

Pour la maximiser, il suffit de la dériver par rapport à chacune des deux variables α_i.

Cette modification va certainement améliorer la précision des résultats donnés par l'algorithme avec regard rétrograde, mais la détermination des deux paramètres α_1 et α_2 en pratique, n'est pas simple.

Une idée déjà mentionnées dans [3] serait de prendre un vecteur b aléatoire et de résoudre les systèmes (2.13) pour avoir une estimation de $\|R^{-1}\|$. Cette idée peut être réutilisée, mais dans le but de construire un vecteur b bien représenté dans la direction du vecteur singulier v_n.

Exprimons le vecteur b suivant la base (v_i), $i = 1, \ldots, n$. On a

$$b = \sum_{i=1}^{n} \alpha_i v_i. \tag{3.4}$$

La solution du système

$$R^T x = b, \tag{3.5}$$

s'exprime alors suivant la base (u_i), $i = 1, \ldots, n$, par

$$x = \sum_{i=1}^{n} (\alpha_i/\sigma_i) u_i. \tag{3.6}$$

Si α_n n'est pas très petit par rapport aux autres composantes α_i, on voit qu'il a suffi de résoudre un seul système linéaire pour produire un vecteur x fort probablement bien représenté dans la direction de u_n. Il est évident que cette probabilité est d'autant meilleure que la matrice R est mal conditionnée, (i.e. $\sigma_1 \gg \sigma_n$).

Maintenant la solution du système

$$Ry = x \tag{3.7}$$

pourra s'exprimer au moyen des v_i par:

$$y = \sum_{i=1}^{n} (\alpha_i/\sigma_i^2) v_i. \tag{3.8}$$

Cette dernière nous laisse voir que même si α_n est petit par rapport aux autres composantes de b, le vecteur y qui est σ_n^{-2} fois mieux représenté que b dans la direction de v_n, a de fortes chances d'être complètement dominé par la composante de v_n.

Finalement, à moins que α_n ne soit nul, on a trouvé un moyen simple pour produire un vecteur y, dominé par sa composante de v_n.

Pour éviter que α_n soit nul, une méthode empirique consiste à partir d'un vecteur b dont les composantes sont générées d'une manière "pseudo-aléatoire" d'une distribution uniforme entre -1 et +1.

Il est alors peu probable qu'on engendre un vecteur b perpendiculaire au vecteur singulier v_n.

Pour les matrices non creuses dont l'ordre n'est pas très petit, même s'il arrive que le vecteur aléatoire b soit numériquement perpendiculaire au vecteur singulier v_n, il ne va pas en général en être de même du vecteur solution y qui est obtenu par la résolution de deux systèmes, du fait que la résolution d'un système ne donne presque jamais une solution exacte.

Il est donc clair qu'il suffit de refaire ce qu'on a déjà fait en commençant cette fois-ci avec le vecteur y de (3.8).

Résolvant les deux systèmes

$$R^T z = y \tag{3.9}$$

et

$$Rt = z, \tag{3.10}$$

on obtient les solutions respectives

$$z = \sum_{i=1}^{n} (\alpha_i/\sigma_i^3) u_i \tag{3.11}$$

et

$$t = \sum_{i=1}^{n} (\alpha_i/\sigma_i^4) v_i. \tag{3.12}$$

On voit facilement l'avantage que nous donne la solution z de (3.9), qui est σ_n^{-2} fois mieux représentée que la solution x de (3.5), dans la direction de u_n.

Maintenant on est presque sûr que le rapport $||t||_2/||z||_2$ donnera une très bonne estimation de $||R^{-1}||_2$ même si le vecteur intial b n'est pas bien représenté dans la

direction du vecteur singulier v_n, car on a

$$\frac{||t||_2}{||z||_2} = \left[\frac{\sum_{i=1}^{n}(\alpha_i/\sigma_i^4)^2}{\sum_{i=1}^{n}(\alpha_i/\sigma_i^3)^2}\right]^{1/2}. \qquad (3.13)$$

Pour vérifier la performance de cette méthode en pratique, on a fait des tests sur 280 matrices triangulaires, dont les ordres varient de 5 à 35, avec 40 matrices de chaque ordre.

Les éléments de celles-ci ainsi que les composantes du vecteur b, ont été pris d'une manière pseudo-aléatoire d'une distribution uniforme entre -1 et +1.

Pour 40 matrices de chaque ordre on a calculé la moyenne des $\hat{\sigma}_n^{-1}/\sigma_n^{-1}$.

n	$\hat{\sigma}_n^{-1}/\sigma_n^{-1}$ moyen
5	0.9998
10	0.9966
15	0.9977
20	0.9997
25	1.0000
30	1.0000
35	0.9999

Table 3.1: Résultats des tests.

Pour le cas d'une matrice triangulaire, le coût de cette méthode est de $2n^2$ flops. Elle nécessite aussi $(n+1)$ cases mémoires additionnelles.

En dehors du fait qu'elle soit la plus économique comparée aux méthodes présentées dans le chapitre précédent, son très haut degré de précision nous laisse voir qu'elle est aussi très fiable.

Parmi les 280 matrices traitées, dans 90 % des cas, le rapport $\hat{\sigma}_n^{-1}/\sigma_n^{-1}$ était supérieur à 0.99. De plus aucun rapport n'a été inférieur à 0.6.

Aucune remise à l'échelle des vecteurs n'a été nécessaire. Toutefois il peut arriver que la matrice R soit presque singulière à cause de certains éléments diagonaux r_{kk} très petits.

Dans ce cas pour éviter un dépassement des capacités de l'ordinateur, une remise à l'échelle, "rescaling", de certains vecteurs est nécessaire.

Afin de ne pas augmenter inutilement le coût de cette méthode il serait préférable de procéder à cette remise à l'échelle en divisant les composantes de ces vecteurs

par des puissances de β où β est la base du système d'arithmétique en point flottant utilisé par l'ordinateur.

On remarque que cette méthode est liée à la méthode des puissances inverses, du fait qu'on a itérer deux fois sur $(R^T R)^{-1}$ avec un b de départ aléatoire. Le haut degré de précision de cette méthode est, bien sûr, en accord avec le fait que le taux de convergence de l'algorithme des puissances inverses reste de très loin meilleur que celui de l'algorithme des puissances directes.

Une méthode probabiliste proposée par J. D. Dixon, qui consiste à prendre un b aléatoire et itérer plus de deux fois sur $(R^T R)^{-1}$ a été décrite dans [5]. Elle donne des bornes inférieure et supérieure pour $||T^{-1}||_2$ avec une probabilité d'au moins 0.99. Son coût est de $3n^2$ à $5n^2$ flops.

Puisqu'un algorithme basé sur un vecteur b aléatoire, qui ne tient pas compte des coefficients de la matrice peut être déplaisant, on peut faire un choix judicieux de b de la manière suivante.

Soit R une matrice inversible d'ordre n.

On a

$$Rb = \sum_{i=1}^{n} b_i r_i, \tag{3.14}$$

où b est un vecteur colonne de $\mathbf{R}^n$ et r_i la i-ème colonne de R.

On peut vérifier que

$$||Rb||_2^2 = \sum_{i=1}^{n} b_i^2 ||r_i||_2^2 + 2 \sum_{i=1}^{n} \sum_{j=i}^{n} b_i b_j ||r_i||_2 ||r_j||_2 \cos \alpha_{ij} , \tag{3.15}$$

où α_{ij} est l'angle entre la i-ème et la j-ème colonne de R.

On sait que

$$\sigma_1^2 = ||R||_2^2 = \max_{||b||_2 = 1} ||Rb||_2^2 \tag{3.16}$$

et

$$\sigma_n^2 = ||R^{-1}||_2^{-2} = \min_{||b||_2 = 1} ||Rb||_2^2. \tag{3.17}$$

On a déjà montré que le maximum en (3.16) sera atteint quand $b = v_1$ et que le minimum en (3.17) sera atteint quand $b = v_n$.

Puisque notre premier but est de produire un vecteur b non perpendiculaire à v_n, l'idée sera de choisir celui-ci de manière à minimiser la fonction (3.15).

La détermination du vecteur b qui minimise la fonction (3.15) coûtera très cher en pratique. Une technique pratique est d'essayer de minimiser (3.15) par rapport à un vecteur b avec des composantes $b_s = \pm 1$.

Si R est une matrice triangulaire inférieure, le schéma est le suivant:

Prenons $b_1 = 1$. La deuxième composante b_2 sera choisie de manière à minimiser la quantité

$$r_{11}^2 + \sum_{i=2}^{n}(r_{i1} + b_2 r_{i2})^2.$$

En général, si $b_1, \ldots, b_{k-1}$ sont déjà calculés, le signe de b_k sera choisi de manière à minimiser

$$\sum_{i=k}^{n}(r_{i1} + b_2 r_{i2} + \cdots + b_k r_{ik})^2. \qquad (3.18)$$

L'algorithme qui découle de cette deuxième méthode est

Algorithme 3

(i) Choisir un vecteur b selon le schéma précédent,

(ii) résoudre les deux systèmes

$$R^T x = b$$

et

$$Ry = x,$$

pour produire un vecteur y, dominé par la composante du vecteur singulier v_n, et enfin

(iii) résoudre une fois de plus les deux systèmes

$$R^T z = y$$

et

$$Rt = z,$$

et prendre $\|t\|_2 / \|z\|_2$, comme estimation de $\|R^{-1}\|_2$.

Cet algorithme coûte $3n^2$ flops.

Des tests faits sur celui-ci ont donné les résultats suivants:

On a essayé 320 matrices triangulaires, dont les ordres varient de 2 à 20 avec 40 de chaque ordre. Les éléments de celles-ci ont été pris d'une distribution uniforme entre -1 et +1.

n	$\hat{\sigma}_n^{-1}/\sigma_n^{-1}$ moyen
2	0.98
3	0.97
4	0.98
5	0.97
6	0.98
10	0.99
15	0.99
20	0.99

Table 3.2: Moyennes des $\hat{\sigma}_n^{-1}/\sigma_n^{-1}$.

De ces 320 matrices testées, aucun rapport n'a été inférieur à 0.7.

On remarque qu'il y a eu une légère diminution dans la précision quand b est choisi selon le schéma précédent. Cela est simplement dû aux restrictions faites sur le choix du vecteur b. Prendre un b aléatoire nous donne de grandes chances pour que ce dernier soit bien représenté dans la direction de v_n. La technique précédente tend en premier lieu à produire un vecteur b non perpendiculaire à v_n.

3.4 Estimation de $||A^{-1}||_2$ à partir de la décomposition LU

Dans la pratique les système linéaires denses sont généralement résolus par l'élimination de Gauss avec une certaine forme de pivotage.

Soit $PA = LU$ la décomposition de Gauss avec pivotage partiel.

Pour des raisons de simplicité écrivons

$$A = LU. \tag{3.19}$$

On doit rappeler que quand on pivote, une quelconque mal condition de A se reflète généralement dans U. La matrice triangulaire inférieure L est en général bien conditionnée. Mais il peut toutefois arriver que L soit très mal conditionnée.

Puisque les facteurs LU produits par la méthode de Gauss ne sont en général pas exacts, un petit problème peut se poser.

Résoudre le premier système

$$A^T x = b \tag{3.20}$$

revient à résoudre les deux systèmes triangulaires

$$U^T y = b, \tag{3.21}$$

$$L^T x = y. \tag{3.22}$$

On doit donc s'attendre à ce que la manipulation des deux matrices L et U cause une baisse dans la précision obtenue dans le cas où la matrice en question est déjà triangulaire.

Fort heureusement cette baisse reste faible.

Des tests dans le cas où le vecteur b est aléatoire ont été faits. Pour 240 matrices pleines dont les ordres varient de 5 à 30 et dont les éléments sont générés entre -1 et +1 d'une manière " pseudo-aléatoire ", on a eu les résultats suivants:

n	$\hat{\sigma}_n^{-1}/\sigma_n^{-1}$ moyen
5	0.97
10	0.96
15	0.95
20	0.97
25	0.95
30	0.93

Table 3.3: Résultats.

De ces 240 matrices on a eu les pourcentages suivants:

>	<=	
0.99	1.00	80 %
0.9	0.99	15 %
0.5	0.9	5 %

Table 3.4: Pourcentages des réussites.

Comme prévu, on voit qu'il y a eu une petite diminution dans la précision de la méthode, mais les résultats restent très satisfaisants sur le plan numérique. Parmis les 240 matrices testées, 80 % des rapports $\hat{\sigma}_n^{-1}/\sigma_n^{-1}$ sont compris entre 0.99 et 1.

3.5 Comparaison des différentes méthodes exposées et conclusion

Le tableau ci-dessous compare les résultats obtenus avec l'algorithme 3 à ceux des méthodes présentées aux chapitre 2, dans le cas d'une matrice triangulaire.

Soient :

E1 : Méthode de Cline, Moler, Stewart et Wilkinson. [p. 23]

E2 : Méthode de Cline, Conn et Van Loan. [p. 28]

E3 : Méthode par l'optimisation convexe de W. W. Hager. [p. 33]

E4 : Méthode probabiliste de J. D. Dixon. [p. 43]

E5 : Méthode proposée (Algorithme 3) de O. Slimani. [p. 44]

Estimation par	Norme	Coût en flops	Fiabilité
E1	1	$5/2n^2$	bonne
E2	2	$5n^2$	bonne
E3	1	$2n^2$ ou 3 n^2 en pratique	bonne
E4	2	$3n^2$ à $5n^2$	bonne
E5	2	$3n^2$	bonne

Table 3.5: Comparaison des méthodes

On conclut ce chapitre avec la remarque que dans tous les cas, l'algorithme 3 donne des résultats assez précis pour être utilisé. Si la matrice est dérivée d'une décomposition QR, on a vu que l'algorithme avec regard rétrograde fournit une très bonne estimation de $||R^{-1}||_2$; mais vu son coût de $5n^2$ flops il peut être préférable d'utiliser la méthode E5 qui ne coûte que $3n^2$ flops.

Un autre cas important où l'algorithme 3 peut être utilisé, est celui des matrices bandes symétriques et définies positives. Car dans ce cas la décomposition de Chokesky qui est plus économique que l'élimination gaussienne est en général la plus utilisée.

Bibliographie

[1] R. Byers, *A LINPACK style condition estimator for the equation AX-XB' = C*, IEEE Trans. Automat. Control. AC-29 (1984), pp. 926-928.

[2] A. K. Cline, A. R. Conn and C. F. Van Loan, *Generalizing the LINPACK condition estimator*, in Numerical Analysis, Mexico 1981, J. P. Hennart, ed., Lecture Notes in Mathematics 909, Springer-Verlag, Berlin, 1982, pp. 73-83.

[3] A. K. Cline, C. B. Moler, G. W. Stewart and J. H. Wilkinson, *An estimate for the condition number of a matrix*. SIAM J. Numer. Anal., 16 (1979), pp. 368-375.

[4] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[5] J. D. Dixon, *Estimating eigenvalues and condition numbers of matrices*. SIAM J. Numer. Anal., 20 (1983), pp. 812-814.

[6] J. J. Dongara, J. R. Bunch, C. B. Moler and G. W. Stewart, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA. 1979.

[7] I. S. Duff, A. M. Erisman and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London. 1986.

[8] G. E. Forsythe and C. B. Moler, *Computer Solutions of Linear Algebraic Equations*, Prentice-Hall Englewood Cliffs, 1967.

[9] D. M. Gay, *A trust-region to linearly constrained optimization*, in Numerical Analysis, Dundee, Scotland, 1983, D. F. Griffiths, ed., Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin,1984, pp. 72-105.

[10] P. E. Gill, W. Murray and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[11] G. H. Golub, S. Nash and C. F. Van Loan, *A Hessenberg-Schur method for the problem AX + XB = C*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 909-913.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[13] R. G. Grimes and J. G. Lewis, *Condition number estimation for sparse matrices*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 384-388.

[14] W. W. Hager, *Condition estimators*, SIAM J Sci. Statist. Comput., 5 (1984), pp. 311-316.

[15] S. J. Hammarling, *Numerical solution of the stable, nonnegative definite Lyapunov equation*, IMA J. Numer. Anal, 2 (1982), pp. 303-323.

[16] N. J. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Review, 4 (1987), pp. 575-596.

[17] N. J. Higham, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405-430.

[18] N. J. Higham, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J, Sci. Statist. Comput., 7 (1986), pp. 150-165.

[19] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[20] T. A. Manteuffel, *An interval analysis approach to rank determination in linear least squares problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 335-348.

[21] C. B. Moler and C. F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Review, 20 (1978), pp. 801-836.

[22] D. P. O'Leary, *Estimating matrix condition numbers*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 205-209.

[23] B. Smith et al. *EISPACK Guide*, Springer-Verlag, New York, 1974.

[24] C. F. Van Loan, *On estimating the condition of eigenvalues and eigenvectors*, Linear Algebra App., 88/89 (1987), pp. 715-732.

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London-Oxford, 1965.