Ann. Sci. Math. Québec, **16** (1), 1992, 95–108.

ESTIMATION DU CONDITIONNEMENT SPECTRAL D'UNE MATRICE

OMAR SLIMANI ET RÉMI VAILLANCOURT

RÉSUMÉ. On présente deux algorithmes, dont un nouveau, pour estimer le conditionnement spectral $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$ d'une matrice régulière triangulaire d'ordre n. Le nombre d'opérations en virgule flottante requis est de l'ordre de $O(n^2)$. On présente les résultats de tests de performance numérique et on fait la comparaison avec d'autres algorithmes connus.

ABSTRACT. Two algorithms, one of which is new, are presented to estimate the spectral condition number, $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$, of a non-singular triangular matrix $A \in R^{n\times n}$. The cost, in flops, of these algorithms is of the order $O(n^2)$. Results of numerical tests are tabulated and comparison is made with existing algorithms. See the English extended abstract at the end of the paper.

1. Introduction. Le conditionnement, $\kappa(A) = \|A\|\,\|A^{-1}\|$, d'une matrice régulière A d'ordre n joue un rôle central dans l'estimation des erreurs de la solution numérique du système linéaire $A\mathbf{x} = \mathbf{b}$ (v. [9], sections 2.5 et 4.5). Il en est de même du conditionnement de la matrice X des vecteurs propres de A, $X^{-1}AX = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$, dans l'estimation de l'erreur sur les valeurs propres λ_i obtenues numériquement (v. [9], section 7.2).

Puisque le conditionnement de A requiert la connaissance de A^{-1} et que le calcul de A^{-1} est de l'ordre de $O(n^3)$ opérations en virgule flottante ("flops", sigle employé pour désigner "floating point operations"), les numériciens ont proposé divers algorithmes, dont le coût est de l'ordre de $O(n^2)$ flops, pour estimer $\|A^{-1}\|$ dans les normes l_1, l_2 et l_∞ (v., par exemple, [2]–[4], [10]–[11], [14]–[16]). Les bibliothèques de programmes (v., par exemple, LINPACK [5] et IMSL [12]) utilisent de tels algorithmes.

La norme vectorielle euclidienne l_2 est la distance naturelle et c'est la plus utilisée en physique. On présente donc deux algorithmes simples, dont un nouveau, obtenus en [18], appelés ici algorithmes 1 et 2 pour estimer le conditionnement spectral $\kappa_2(A)$ avec une grande précision et dont le coût est de l'ordre de $O(n^2)$ flops. On vérifie leur performance au moyen de tests numériques sur MATLAB et on la compare avec celle d'algorithmes existants.

2. Le conditionnement d'une matrice. Soient un vecteur $\mathbf{x} \in R^n$ et une matrice $A \in R^{n\times n}$. Pour $\mathbf{x} \in C^n$ et $A \in C^{n\times n}$, on remplacera les transposés $\mathbf{x}^T$ et A^T par le

Reçu le 22 août 1991 et, sous forme définitive, le 19 mars 1992.

Avec le concours du Conseil de recherche en sciences naturelles et en génie du Canada, octroi n° A 7691, et du Centre de recherches mathématiques de l'Université de Montréal

conjugué complexe de leurs transposés $\mathbf{x}^H$ et A^H et matrice orthogonale par matrice unitaire. On définit la norme l_p, $1 \leq p \leq \infty$, de $\mathbf{x}$ par

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}, \qquad 1 \leq p < \infty, \tag{2.1}$$

et

$$\|\mathbf{x}\|_\infty = \max\{|x_1|, |x_1|, \ldots, |x_n|\}. \tag{2.2}$$

La p-norme de A subordonnée à celle de $\mathbf{x}$ est définie par

$$\|A\|_p = \sup_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}, \tag{2.3}$$

et si A est régulière, on peut définir la p-norme de son inverse A^{-1} par

$$\|A^{-1}\|_p^{-1} = \inf_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}. \tag{2.4}$$

Définition 2.1. Le conditionnement d'une matrice régulière A est défini dans la p-norme par

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p, \qquad 1 \leq p \leq \infty. \tag{2.5}$$

On appelle conditionnement spectral de A le nombre $\kappa_2(A)$.

L'appellation conditionnement spectral est prise de Wilkinson [19]. Voici trois caractérisations remarquables du conditionnement de A.

(a) $\kappa_p(A)$ mesure l'inverse de la distance relative, dans la p-norme, de A au sous-espace des matrices singulières (v. Kahan [13]):

$$\frac{1}{\kappa_p(A)} = \min_{A+E \text{ sing}} \frac{\|E\|_p}{\|A\|_p}. \tag{2.6}$$

(b) $\kappa(A)$ est la dérivée de Fréchet normalisée de l'application $A \to A^{-1}$ dans une norme matricielle quelconque (v. Rice [17]):

$$\kappa(A) = \lim_{\delta \to 0} \sup_{\|E\| \leq \delta \|A\|} \frac{\|(A+E)^{-1} - A^{-1}\|}{\delta \|A^{-1}\|}. \tag{2.7}$$

(c) Soit θ l'angle minimum entre $A\mathbf{x}$ et $A\mathbf{y}$, toute paire $\{\mathbf{x}, \mathbf{y}\}$ orthogonale. Alors

$$\kappa_2(A) = \cot \frac{\theta}{2}. \tag{2.8}$$

Ce résultat découle de l'inégalité de Wielandt [20]:

$$\frac{|\mathbf{x}^T A^T A \mathbf{y}|^2}{(\mathbf{x}^T A^T A \mathbf{x})(\mathbf{y}^T A^T A \mathbf{y})} \leq \cos^2 \theta. \tag{2.9}$$

Le conditionnement d'une matrice occupe une place centrale dans le calcul matriciel. En voici des exemples.

(1) Soit A régulière et considérons le système $A\mathbf{x} = \mathbf{b}$. L'erreur relative de la solution du système perturbé

$$(A+F)(\mathbf{x}+\mathbf{w}) = \mathbf{b}+\mathbf{f}, \qquad \|A^{-1}\|_p\|F\|_p < 1, \tag{2.10}$$

satisfait l'inégalité (v. [15]):

$$\frac{\|\mathbf{w}\|_p}{\|\mathbf{x}\|_p} \leq \frac{\kappa_p(A)}{1-\kappa_p(A)\frac{\|F\|_p}{\|A\|_p}}\left(\frac{\|\mathbf{f}\|_p}{\|\mathbf{b}\|_p}+\frac{\|F\|_p}{\|A\|_p}\right). \tag{2.11}$$

(2) Si $F = 0$ dans (2.10), l'inégalité précédente se réduit à

$$\frac{\|\mathbf{w}\|_p}{\|\mathbf{x}\|_p} \leq \kappa_p(A)\frac{\|\mathbf{f}\|_p}{\|\mathbf{b}\|_p}. \tag{2.12}$$

(3) Soit $\hat{\mathbf{x}}$ la solution numérique de $A\mathbf{x} = \mathbf{b}$ obtenue par calcul en virgule flottante en base β avec partie fractionnaire à t chiffres. Les numériciens (v. [9]) considèrent que l'erreur relative en $\hat{\mathbf{x}}$ satisfait l'estimation suivante:

$$\frac{\|\hat{\mathbf{x}}-\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \approx \beta^{-t}\kappa_p(A). \tag{2.13}$$

Il suit de cette estimation que si $\beta^{-t}\kappa_p(A) \approx 1$, $\hat{\mathbf{x}}$ n'a même pas un seul chiffre correct.

(4) La résolution numérique de $A\mathbf{x} = \mathbf{b}$ par l'élimination gaussienne avec pivotage partiel produit la décomposition $PA = LU$, P une permutation de l'identité, L triangulaire inférieure aux éléments $|l_{ij}| \leq 1$ et U triangulaire supérieure. Alors la solution numérique $\hat{\mathbf{x}}$ s'obtient au moyen des substitutions directe et rétrograde:

$$L\mathbf{y} = P\mathbf{b}, \qquad U\mathbf{x} = \mathbf{y}. \tag{2.14}$$

On améliore $\hat{\mathbf{x}} = \mathbf{x}_0$ par la récurrence suivante:

$$\left.\begin{array}{ll} \mathbf{r} = \mathbf{b} - A\mathbf{x}_0 & \text{calcul du résidu en précision double} \\ L\mathbf{y} = P\mathbf{r} & \text{résolution par substitution directe} \\ U\mathbf{z} = \mathbf{y} & \text{résolution par substitution rétrograde} \\ \mathbf{x}_1 = \mathbf{x}_0 + \mathbf{z} & \text{valeur améliorée} \end{array}\right\} \tag{2.15}$$

Les numériciens acceptent la règle pratique suivante:
Règle pratique. Soit $u = \beta^{-d}$ la précision de la machine, c'est-à-dire $1+\delta = 1$ tout $|\delta| < u$. Si $\kappa(A) \approx \beta^q$, alors le nombre de chiffres corrects de $\mathbf{x}_k$ obtenus par la récurrence (2.15) est approximativement

$$\min\{d, k(d-q)\}. \tag{2.16}$$

Si $d > q$, on voit que plus d est grand et plus $\kappa(A)$ est petit, meilleure sera la solution numérique.

(5) Le conditionnement de la matrice des vecteurs propres d'une matrice diagonalisable contrôle la perturbation des valeurs propres (v. [20]). On a, par exemple, le théorème suivant.

Théorème 2.1 (Bauer-Fike [1]). *Soit A diagonalisable:*

$$X^{-1}AX = \operatorname{diag}(\lambda_1, \ldots, \lambda_n),$$

et soit μ une valeur propre de $A + E$. Alors

$$\min_{i=1,\ldots,n} |\lambda_i - \mu| \leq \kappa_p(X)\|E\|_p.$$

Il y a donc une valeur propre de A à une distance d'au plus $\kappa_p(X)\|E\|_p$ d'une valeur propre de $A + E$.

Dans ce travail, on s'intéressera en particulier au cas où $p = 2$ et on parlera du conditionnement spectral de A, $\kappa_2(A)$.

Les numériciens estiment ordinairement le coût d'un algorithme par le nombre d'opérations en virgule flottante ("flop", floating point operation). Puisque le coût du calcul direct de $\kappa_2(A)$ peut être élevé, en pratique de l'ordre de $O(n^3)$ flops, on se contentera d'une estimation qui peut être obtenue avec $O(n^2)$ opérations arithmétiques.

Il est clair que la difficulté dans le calcul de $\|A\|_2\|A^{-1}\|_2$ provient du facteur $\|A^{-1}\|_2$, du fait que le calcul de A^{-1} est de l'ordre de $O(n^3)$ flops. En effet, la norme $\|A\|_2$ peut être estimée par l'une des inégalités suivantes:

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2, \qquad \frac{1}{\sqrt{n}}\|A\|_p \leq \|A\|_2 \leq \sqrt{n}\|A\|_p, \quad p = 1, \infty, \tag{2.17}$$

où $\|A\|_F$ est la norme de Frobenius ou norme matricielle euclidienne de A,

$$\|A\|_F^2 = \sum_{i,j=1}^{n} |a_{ij}|^2. \tag{2.18}$$

Dans la pratique, on obtient généralement une factorisation de A sous une forme contenant un facteur triangulaire, par exemple QR ou LU. Puisque Q est orthogonale ou unitaire et L est ordinairement bien conditionnée, il suffit d'estimer le conditionnement du facteur triangulaire R ou U. Par conséquent, le problème se résume à estimer $\|A^{-1}\|_2$ quand A est triangulaire.

3. La décomposition selon les valeurs singulières. Avant de formuler les algorithmes 1 et 2 pour estimer $\|A^{-1}\|_2$, on analyse le problème en termes de la décomposition en valeurs singulières (v. [3]).

Il est connu (v. [8], section 4, ou [9], section 2.6) que A admet une décomposition selon ses valeurs singulières $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$:

$$A = U\Sigma V^T, \tag{3.1}$$

où $U = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ et $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ sont orthogonales et $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$. A est régulière si et seulement si $\sigma_n > 0$.

Il découle de (3.1) que

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i \tag{3.2}$$

et

$$A^T\mathbf{u}_i = \sigma_i \mathbf{v}_i. \tag{3.3}$$

Du fait que U et V sont orthogonales et Σ diagonale, on sait que

$$\|A\|_2 = \|U\Sigma V^T\|_2 = \|\Sigma\|_2 = \sigma_1 \tag{3.4}$$

et si A est régulière,

$$\|A^{-1}\|_2 = \|V\Sigma^{-1}U^T\|_2 = \|\Sigma^{-1}\|_2 = \frac{1}{\sigma_n}. \tag{3.5}$$

Alors

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}. \tag{3.6}$$

On peut calculer σ_n au moyen d'une modification de la méthode de la puissance inverse énoncée au théorème suivant.

Théorème 3.1. *Soient le vecteur initial*

$$\mathbf{y}_0 = \beta_1\mathbf{v}_1 + \cdots + \beta_n\mathbf{v}_n, \qquad \beta_n \neq 0, \tag{3.7}$$

et la récurrence

$$\left.\begin{aligned} A^T\mathbf{x}_m &= \mathbf{y}_{m-1} \\ A\mathbf{y}_m &= \mathbf{x}_m \end{aligned}\right\}, \qquad m = 1, 2, \ldots . \tag{3.8}$$

Alors

$$\lim_{m\to\infty} \frac{\|\mathbf{y}_m\|_2}{\|\mathbf{x}_m\|_2} = \frac{1}{\sigma_n}. \tag{3.9}$$

Démonstration. Par (3.3) et (3.2)

$$\mathbf{x}_m = \sum_{i=1}^{n} \frac{\beta_i}{\sigma_i^{2m-1}}\mathbf{u}_i \tag{3.10}$$

et

$$\mathbf{y}_m = \sum_{i=1}^{n} \frac{\beta_i}{\sigma_i^{2m}}\mathbf{v}_i. \tag{3.11}$$

Puisque $\sigma_n \leq \sigma_j$, $j = n-1, \ldots, 1$, et $\beta_n \neq 0$, le résultat suit des approximations asymptotiques

$$\|\mathbf{x}_m\| \approx \frac{|\beta_n|}{\sigma_n^{2m-1}} \tag{3.12}$$

et

$$\|\mathbf{y}_m\| \approx \frac{|\beta_n|}{\sigma_n^{2m}} \tag{3.13}$$

quand $m \to \infty$. □

Dans le cas où A est triangulaire, le coût d'une itération de la récurrence (3.8) est de l'ordre de $O(n^2)$ flops. On aura alors une méthode $O(n^2)$ pour estimer $\|A^{-1}\|_2$ si l'on sait choisir $\mathbf{y}_0$ avec $\beta_n \neq 0$ d'une façon économique en flops. Deux choix empiriques se présentent:

(a) Choisir $\mathbf{y}_0$ d'une façon aléatoire.
(b) Prendre $\mathbf{y}_0 = (1, \pm 1, \pm 1, \ldots, \pm 1)^T$ pour maximiser β_n devant β_j, $j = 1, \ldots, n-1$.

Ces choix seront l'objet des deux sections suivantes et donneront lieu à deux méthodes qu'on appellera algorithmes 1 et 2.

4. L'algorithme 1. Le premier algorithme utilise un vecteur de départ $\mathbf{y}_0 = \mathbf{b} = (b_1, \ldots, b_n)^T$ aléatoire et deux itérations de la récurrence (3.8) pour estimer σ_n par $\hat{\sigma}_n$:

$$\frac{1}{\hat{\sigma}_n} = \left[\frac{\sum_{i=1}^n (\beta_i/\sigma_i^4)^2}{\sum_{i=1}^n (\beta_i/\sigma_i^3)^2}\right]^{1/2} \approx \frac{1}{\sigma_n}, \tag{4.1}$$

où $\mathbf{b} = \beta_1 \mathbf{v}_1 + \cdots + \beta_n \mathbf{v}_n$.

Algorithme 1. *Pour estimer σ_n par $\hat{\sigma}_n$:*

(1) *Choisir les composantes de* $\mathbf{b}$ *d'une manière pseudo-aléatoire d'une distribution uniforme entre* -1 *et* $+1$.
(2) *Résoudre:*

$$A^T\mathbf{x} = \mathbf{b}, \qquad A\mathbf{y} = \mathbf{x}. \tag{4.2}$$

(3) *Résoudre:*

$$A^T\mathbf{z} = \mathbf{y}, \qquad A\mathbf{t} = \mathbf{z}. \tag{4.3}$$

(4) *Calculer l'estimation* $1/\hat{\sigma}_n$:

$$\frac{1}{\hat{\sigma}_n} = \frac{\|\mathbf{t}\|_2}{\|\mathbf{z}\|_2}. \tag{4.4}$$

Le coût de cette méthode pour A triangulaire est de $2n^2$ flops. Elle nécessite $(n+1)$ cases mémoires additionnelles.

Pour vérifier la performance de l'algorithme 1, on a fait des tests sur 280 matrices triangulaires, dont les ordres variaient de 5 à 35, avec 40 matrices de chaque ordre. Les résultats sont présentés au tableau 4.1.

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.

En plus du fait que cet algorithme soit très économique, son haut degré de précision nous laisse voir qu'il est aussi très fiable. La nouveauté de l'algorithme 1 consiste dans l'ajout de la seconde itération (4.3) à l'algorithme décrit dans [3] en vue d'en augmenter la fiabilité et la précision en prenant $m = 2$ dans le théorème 3.1.

Enfin, considérons l'estimation de $\|A^{-1}\|_2$ d'une matrice pleine à partir de sa décomposition LU. Les systèmes linéaires denses sont généralement résolus par l'élimination gaussienne avec un pivotage partiel sur les lignes. Soit

$$PA = LU \tag{4.5}$$

TABLEAU 4.1. Pour 40 matrices triangulaires d'ordre n, moyenne du rapport indiqué pour l'algorithme 1.

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

TABLEAU 4.2. Pour 240 matrices pleines d'ordre n, moyenne du rapport indiqué pour l'algorithme 1.

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

la décomposition de Gauss avec pivotage partiel décrite à l'exemple (4) de la section 2. Le pivotage rejette ordinairement une quelconque mal condition de A sur U. La matrice triangulaire inférieure L est en général bien conditionnée; mais il peut toutefois arriver qu'elle soit très mal conditionnée.

Puisque les facteurs L et U produits par la méthode de Gauss en général ne sont pas exacts, il s'ensuit une perturbation de la solution numérique qui satisfait l'inégalité (2.11), d'où l'importance d'une estimation du conditionnement de A.

Lors de l'application de l'algorithme 1, on remplace A par LU et on résout les systèmes obtenus par substitutions directe et rétrograde comme en (2.14) ou (2.15). La manipulation de deux matrices triangulaires, L et U, au lieu d'une seule, peut réduire la précision de l'estimation de $\|A^{-1}\|$. Fort heureusement dans la plupart des cas cette baisse reste faible.

On a fait des tests sur des matrices pleines avec l'algorithme 1 et un vecteur aléatoire b. Pour 240 matrices denses 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 obtenu les résultats présentés au tableau 4.2.

Les pourcentages de ces rapports sont présentés au tableau 4.3 pour les intervalles indiqués.

Comme prévu, on voit qu'il y a eu une légère diminution dans la précision de la méthode, mais les résultats restent très satisfaisants sur le plan numérique.

5. Le nouvel algorithme 2. Le nouvel algorithme 2 que nous proposons, à la différence

TABLEAU 4.3. Pourcentage des rapports dans chacun des intervalles $]p, q]$ pour l'algorithme 1 sur LU.

Intervalle	$\hat{\sigma}_n^{-1}$ estimé/σ_n^{-1} exact
$]0.99, 1.00]$	80%
$]0.90, 0.99]$	15%
$]0.50, 0.90]$	5%

de l'algorithme 1, choisit les composantes b_i du vecteur de départ $\mathbf{b} = \mathbf{y}_0$ égales à ± 1.

Soient les matrices A et A_k et les vecteurs $\mathbf{b}$ et $\mathbf{b}_k$ d'ordre respectif n et k et leur produit:

$$A\mathbf{b} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & & 0 \\ \vdots & & \ddots & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, A_k\mathbf{b}_k = \begin{bmatrix} a_{k1} & a_{k2} & \cdots & a_{kk} \\ a_{k+1,1} & a_{k+1,2} & & a_{k+1,k} \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & & a_{nk} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}.$$

Puisque

$$\sigma_n^2 = \|A^{-1}\|_2^{-2} = \min_{\mathbf{b}\neq 0} \frac{\|A\mathbf{b}\|_2^2}{\|\mathbf{b}\|_2^2}, \tag{5.1}$$

l'algorithme 2 choisit la première composante de $\mathbf{b}$ égale à $+1$ et les autres successivement égales à ± 1 de manière à minimiser $\|A_k\mathbf{b}_k\|_2$, c'est-à-dire: on prend b_k tel que

$$\min_{b_k=\pm 1} \|A_k\mathbf{b}_k\|_2, \qquad k = 2, 3, \ldots, n. \tag{5.2}$$

On a donc l'algorithme suivant.

Algorithme 2. *Pour estimer σ_n par $\hat{\sigma}_n$:*

(1) *Choisir $b_1 = +1$. Si $b_1, \ldots, b_{k-1}$ sont déjà calculés, le signe de $b_k = \pm 1$ sera celui qui minimise*

$$\sum_{i=k}^{n} (a_{i1} + a_{i2}b_2 + \cdots + a_{ik}b_k)^2. \tag{5.3}$$

(2) *Résoudre:*

$$A^T\mathbf{x} = \mathbf{b}, \qquad A\mathbf{y} = \mathbf{x}. \tag{5.4}$$

(3) *Résoudre:*

$$A^T\mathbf{z} = \mathbf{y}, \qquad A\mathbf{t} = \mathbf{z}. \tag{5.5}$$

(4) *Calculer l'estimation $1/\hat{\sigma}_n$:*

$$\frac{1}{\hat{\sigma}_n} = \frac{\|\mathbf{t}\|_2}{\|\mathbf{z}\|_2}. \tag{5.6}$$

TABLEAU 5.1. Pour 40 matrices d'ordre n, moyenne du rapport indiqué pour l'algorithme 2.

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

Cet algorithme coûte $3n^2$ flops. On a effectué des tests de performance sur cet algorithme. Pour 320 matrices triangulaires dont les ordres varient de 2 à 20, avec 40 matrices de chaque ordre, et dont les éléments proviennent d'une distribution uniforme entre -1 et $+1$, on a obtenu les résultats présentés au tableau 5.1.

Aucun rapport n'a été inférieur à 0.7. De plus, on remarque que l'algorithme est stable dans la précision du rapport

$$\hat{\sigma}_n^{-1} \text{ estimé}/\sigma_n^{-1} \text{ exact}$$

quand la dimension de la matrice générée varie.

6. Algorithmes de Cline *et al.* La méthode de Cline *et al.* [3] choisit le vecteur de départ $\mathbf{b}$ de telle manière que la solution de

$$A^T\mathbf{x} = \mathbf{b} \tag{6.1}$$

donne un rapport

$$\frac{\|\mathbf{x}\|_1}{\|\mathbf{b}\|_1} \quad (\approx \|A^{-T}\|_1) \tag{6.2}$$

aussi grand que possible. Puis elle résout

$$A\mathbf{y} = \mathbf{x} \tag{6.3}$$

et estime $\|A^{-1}\|$ par le rapport

$$\frac{\|\mathbf{y}\|_1}{\|\mathbf{x}\|_1} \quad (\approx \|A^{-1}\|_1). \tag{6.4}$$

Pour motiver l'étape (6.3)–(6.4), on considère que si le rapport (6.2) est grand, alors le rapport (6.4) est au moins aussi grand.

O'Leary [14] a modifié cette méthode en combinant (6.2) et (6.4) pour estimer $\|A^{-1}\|_1$ par

$$\max\left\{\frac{\|\mathbf{x}\|_\infty}{\|\mathbf{b}\|_\infty}, \frac{\|\mathbf{y}\|_1}{\|\mathbf{x}\|_1}\right\} (\approx \|A^{-1}\|_1), \tag{6.5}$$

TABLEAU 6.1. Pour 100 matrices d'ordre variant de 5 à 50, r moyen pour l'algorithme 3.

Ordre	r moyen
5	0.69
10	0.60
20	0.52
40	0.43

TABLEAU 6.2. Pourcentage des rapports dans chacun des intervalles $]p, q]$ pour l'algorithme 4 avec regard rétrograde.

Intervalle	$\hat{\sigma}_n^{-1}$ estimé/σ_n^{-1} exact
$]0.9, 1.0]$	62.6%
$]0.8, 0.9]$	11.6%
$]0.7, 0.8]$	6.8%
$]0.6, 0.7]$	3.5%
$]0.5, 0.6]$	4.4%
$]0.0, 0.5]$	10.6%

puisque

$$\|A^{-1}\|_1 = \|A^{-T}\|_\infty \approx \frac{\|\mathbf{x}\|_\infty}{\|\mathbf{b}\|_\infty}. \tag{6.6}$$

Appelons **algorithme 3** l'algorithme qui en découle. Son coût est de $5/2\,n^2$ flops. Les résultats des tests de performance sur 100 matrices de dimensions variant de 5 à 50 dont les éléments furent choisis d'une distribution uniforme entre -1 et $+1$, ont donné pour chaque ordre les moyennes de

$$r = \frac{\|A^{-1}\|_1 \text{ estimé}}{\|A^{-1}\|_1 \text{ exact}} \tag{6.7}$$

qui sont présentées au tableau 6.1.

Un peu plus tard, Cline *et al.* [2] ont généralisé cette méthode au moyen d'une technique de regard rétrograde ("look-behind") qui, une fois les signes des premières composantes de $\mathbf{b}$ choisis, permet de les changer au besoin. On appellera cette méthode **algorithme 4.**

En [2] on a calculé le rapport $\hat{\sigma}_n^{-1}$ estimé/σ_n^{-1} exact pour 1000 matrices triangulaires dont les ordres varient de 5 à 50, avec 100 de chaque ordre, et dont les éléments furent générés aléatoirement entre -1 et $+1$. Les pourcentages de ces rapports sont présentés au tableau 6.2 pour les intervalles indiqués.

7. Algorithme de Hager pour estimer $\|A^{-1}\|_1$. Pour estimer $\|A^{-1}\|_1$, Hager [10] a construit une méthode basée sur l'optimisation convexe: si $B \in R^{n\times n}$, alors

$$f(\mathbf{x}) = \|B\mathbf{x}\|_1 = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} b_{ij} x_j \right|, \tag{7.1}$$

TABLEAU 7.1. Pour 200 matrices d'ordre n, moyenne du rapport indiqué pour l'algorithme de Hager.

n	$\|A^{-1}\|_1$ estimé/$\|A^{-1}\|_1$ exact
5	0.96
10	0.97
20	0.98
40	0.97
80	0.98

est une fonction convexe sur l'ensemble convexe:

$$\mathcal{S} = \{\mathbf{x} \in R^n;\ \|\mathbf{x}\|_1 \leq 1\}, \tag{7.2}$$

et f atteint son maximum sur l'un des $2n$ sommets, $\pm\mathbf{e}_j$, $j = 1, \ldots, n$, de $\mathcal{S}$.

L'algorithme qui en découle exige de $2n^2$ à $3n^2$ flops. On présente au tableau 7.1 le résultat des tests de performance de cette méthode sur plusieurs ensembles de 200 matrices de même dimension, dont les éléments furent pris d'une distribution uniforme sur l'intervalle $[-1, +1]$.

La précision de l'algorithme de Hager dépasse celle des autres algorithmes pour la norme l_1. On remarque aussi que la précision est uniforme par rapport à la dimension de la matrice générée.

8. Conclusion. L'algorithme 1 et le nouvel algorithme 2 restent très compétitifs devant les algorithmes [3]–[4], [10]–[11] et [14], vu leur précision élevée, la facilité de leur mise en œuvre et l'indépendance de leur précision vis-à-vis de la dimension de la matrice. Etant donné l'intérêt spécial que l'on peut avoir pour la norme l_2, on aura intérêt à les mettre en œuvre et à les utiliser pour estimer le conditionnement spectral d'une matrice.

La bonne performance des méthodes d'évaluation du conditionnement d'une matrice exposées dans ce travail nous porte à croire qu'elles sont fiables dans la pratique. Les considérations suivantes peuvent expliquer cette fiabilité.

1° L'égalité dans (2.12) n'est atteinte que dans le cas extrême où $\mathbf{b} = \beta_1 \mathbf{u}_1$ et $\mathbf{f} = \beta_n \mathbf{u}_n$, ce qui est un cas fort improbable. D'ailleurs (2.12) suit de l'inégalité $\|\mathbf{b}\| \leq \|A\|\,\|\mathbf{x}\|$ qui est très faible pour presque tout $\mathbf{b}$ lorsque $\kappa(A)$ est grand (v. [3]). Donc une sous-estimation de $\|A^{-1}\|$ du type (6.2) ou (6.4) suffit souvent dans la pratique.

2° La récurrence (3.8) donne de meilleurs résultats quand $\sigma_1 \gg \sigma_n$, c'est-à-dire si $\kappa_2(A)$ est grand. Or les tests de fiabilité rapportés ici ont été faits sur des matrices aléatoires, qui sont presque toutes bien conditionnées d'après les résultats de Edelman [6–7]. Donc dans la pratique, une matrice mal conditionnée aura toutes les chances d'être dépistée par l'une des estimations de $\|A^{-1}\|$ présentées dans ce travail.

9. Remerciements. Les auteurs remercie le rapporteur pour ses remarques judicieuses qui ont servi à améliorer le fond et la forme de cet article.

English extended abstract. The condition number, $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$, $p = 1, 2, \infty$, of a nonsingular $n \times n$ matrix A plays a central role in estimating the relative error, (2.11) and (2.12), in the numerical solution to a linear system, $A\mathbf{x} = \mathbf{b}$. Similarly, if the matrix of eigenvectors, X, diagonalizes A, i.e. $X^{-1}AX = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$, and μ is an eigenvalue of the perturbed matrix $A + E$, then, for example, by a theorem of Bauer and Fike [1] (see Theorem 2.1) there is a λ_i such that $|\mu - \lambda_i| \leq \kappa_p(X)\|E\|_p$. It is thus important, in matrix computations, to have an estimate of $\kappa_p(A)$. Since $\|A\|_p$ is easily computed for $p = 1$ and ∞, and $\|A\|_2$ can be estimated by means of the inequalities (2.17), the difficulty in estimating $\kappa_p(A)$ is in estimating $\|A^{-1}\|$. Since the cost of computing A^{-1} is of the order $O(n^3)$ floating point operations ("flops"), numerical analysts have constructed algorithms which estimate $\|A^{-1}\|_p$ in $O(n^2)$ flops. Such algorithms have been incorporated in program libraries, like LINPACK [5] and IMSL [12].

Since in the applications, A has often been factored as the product of a full or triangular matrix (which is usually well conditioned) and a triangular matrix, several $O(n^2)$ methods have been obtained to estimate $\kappa_p(A)$ for a triangular matrix A. Examples of such factorizations are $A = QR$, where Q is orthogonal, and $PA = LU$, where P is a permutation of the identity matrix and L is lower triangular with all elements l_{ij} bounded by one in absolute value.

In this work, we present Algorithm 1, which is an extension of ideas found in [3], and the new Algorithm 2 to estimate the spectral condition number $\kappa_2(A)$ of a triangular matrix A in $O(n^2)$ flops (see [18]).

In Theoroem 3.1, the singular value decomposition (3.1), $A = U\Sigma V^T$, of A is used to construct the iterative scheme (3.7)–(3.9) which converges to the smallest singular value $\sigma_n = \|A^{-1}\|_1^{-1}$ of A, provided the initial vector y_0 is not orthogonal to the n-th column, $\mathbf{v}_n$, of the orthogonal matrix V.

Algorithm 1 uses a random initial vector $\mathbf{y}_0 = \mathbf{b}$, with components b_i, $-1 \leq b_i \leq 1$, taken from a uniform distribution, and uses two iterations in Theorem 3.1, i.e. with $m = 1$ and 2, to estimate $1/\sigma_n$ by $1/\hat{\sigma}_n$ (see (4.2)–(4.4)). This algorithm was tested numerically on sets of 40 random triangular matrices of dimensions 5(5)35 and the averages of $\hat{\sigma}_n/\sigma_n$ are listed in Table 4.1. In 90% of the 280 matrices tested, $\hat{\sigma}_n^{-1}/\sigma_n^{-1} \geq 0.99$ and no ratio was smaller than 0.6. This algorithm is reliable and economical. Results that are almost as good are listed in Table 4.2 and were obtained with Algorithm 1 when applied to dense matrices $PA = LU$.

The new Algorithm 2 uses an intial vector $\mathbf{y}_0 = \mathbf{b} = (1, \pm 1, \ldots, \pm 1)^T$ which minimizes $\|A\mathbf{b}\|_2$ (see 5.1) according to the following scheme: Set $b_1 = +1$; if $b_1, \ldots, b_{k-1}$ have already been picked, choose the sign of $b_k = \pm 1$ in order to minimize (5.3), for $k = 2, \ldots, n$. Then use two iterations (3.8)–(3.9) as in Theorem 3.1. Results of numerical testing are listed in Table 5.1. No ratio of the estimated $\hat{\sigma}_n^{-1}$ to the exact σ_n^{-1} was smaller than 0.7 for 320 random triangular matrices tested. The algorithm is stable with respect to the dimension n of the matrices and its cost is $3n^2$ flops.

Comparative numerical results are listed for the "look-ahead" Algorithm 3 of Cline *et al.* [3] and O'Leary [14], the "look-behind" Algorithm 4 of Cline *et al.* [2], and the l_1-convex optimization Algorithm of Hager [10], in Tables 6.1, 6.2, and 7.1, respectively.

It was found that Algorithms 1 and 2 compete favourably with existing algorithms. Since the l_2 vector norm is widely used, it seems desirable that codes be written for these.

Since the five algorithms reported here perform well on random matrices, it is considered that they are quite reliable in practice. The user's confidence could be partially explained by the following two considerations: (a) Inequality in (2.12) holds only in the extreme case where $\mathbf{b} = \beta_1 \mathbf{u}_1$ and $\mathbf{f} = \beta_n \mathbf{u}_n$. The probability that such a case occcurs in practice is nil (see [3]). Therefore lower estimates of the type (6.2) or (6.4) for $\|A^{-1}\|$ often suffice in practice. (b) The iteration (3.8) gives better results when $\sigma_1 \gg \sigma_n$, that is when $\kappa_2(A)$ is large. Because of results obtained by Adelman [6-7] to the effect that random matrices are well conditioned to a very high probability, one is almost sure that a badly conditioned matrix will be detected by any one of the above five methods, although rare counter-examples could be constructed for any one of the methods.

BIBLIOGRAPHIE

1. F. L. Bauer et C. T. Fike, *Norms and exclusion theorems*, Numer. Math. **2** (1960), 137–141.
2. A. K. Cline, A. R. Conn et C. F. Van Loan, *Generalizing the LINPACK condition estimator*, 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 et J. H. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal. **16** (1979), 368–375.
4. J. D. Dixon, *Estimating eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal. **20** (1983), 812–814.
5. J. J. Dongara, J. R, Bunch, C. B. Moler et G. W. Stewart, *LINPACK Users' Guide*, SIAM, Philadelphia, PA, 1979, ch. 6.
6. Alan Edelman, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl. **9** (1988), 543–560.
7. Alan Edelman, *Eigenvalues and Condition Numbers of Random Matrices*, Dissertation doctorale, Massachussets Intitute of Technology, Cambridge, Mass., 1989, 103 pp.
8. G. E. Forsythe et C. B. Moler, *Computer Solutions of Linear Algebraic Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1967, sect. 3.
9. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983, sections 2.5, 2.6, 4.5 et 7.2.
10. W. W. Hager, *Condition estimators*, SIAM J Sci. Statist. Comput. **5** (1984), 311–316.
11. N. J. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Review **4** (1987), 575–596.
12. IMSL, *IMSL Math/Library User's Manual*, Softcover Ed. 1.1, IMSL, Houston, Texas, 1989.
13. W. M. Kahan, *Numerical linear algebra*, Can. Math. Bull. **9** (1966), 757–801.
14. D. F O'Leary, *Estimating matrix condition numbers*, SIAM J Sci. Statist. Comput. **1** (1980), 205–209.
15. B. Noble et J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1977, sect. 5.4.
16. Wladimir Popov, *An easily computed upper bound for the condition number of a matrix*, Serdica **15** (1989), 192–196.
17. J. Rice, *A theory of condition*, SIAM J. Num. Anal. **3** (1966), 287–310.
18. Omar Slimani, *Estimation du conditionnement d'une matrice*, Thèse de M.Sc., Université d'Ottawa, Ottawa, Canada K1N 6N5, 1989, 52 pp.
19. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965, ch. 2.
20. H. Wielandt, *Inclusion theorems for eigenvalues*, Nat. Bur. Standards Appl. Math. Ser. **29** (1953), 75–78.

OMAR SLIMANI
DÉPARTEMENT DE MATHÉMATIQUES
UNIVERSITÉ DE BÉZIA
BÉZIA, ALGÉRIE

Rémi Vaillancourt
Département de mathématiques
Université d'Ottawa
Ottawa, ON, Canada K1N 6N5