Calcul matriciel

Notes[1] pour les cours MAT 5580 et 4781

Rémi VAILLANCOURT

Département de mathématiques et de statistique
Université d'Ottawa
Ottawa, ON, Canada K1N 6N5

le 03 novembre 1995

[1]Avec le concours du Conseil de l'éducation franco-ontarien.

Table des matières

1 Eléments d'algèbre des matrices **1**
- 1.1 Vecteurs et matrices . . . 1
- 1.2 Indépendance, orthogonalité, sous-espace . . . 1
- 1.3 Matrices spéciales . . . 2
- 1.4 Matrices blocs . . . 3

2 Normes et valeurs singulières **5**
- 2.1 Normes de vecteurs . . . 5
- 2.2 Normes de matrices . . . 7
- 2.3 Décomposition en valeurs singulières . . . 8
- 2.4 Quantification de la presque déficience . . . 12
- 2.5 Conditionnement des systèmes carrés . . . 13

3 Calculs en virgule flottante **17**
- 3.1 Nombres en virgule flottante . . . 17
- 3.2 Ordinateur modèle . . . 19
- 3.3 Nombre de chiffres significatifs . . . 22
- 3.4 Effet cumulatif des erreurs . . . 23
- 3.5 Erreurs d'arrondi des calculs matriciels . . . 27
- 3.6 Analyse de l'erreur rétrograde . . . 28
- 3.7 Le calcul de la norme $\|x\|_2$ . . . 29

4 Décomposition des matrices en facteurs **31**
- 4.1 Réflexions de Householder . . . 31
- 4.2 Rotations de Givens . . . 37
- 4.3 Transformations de Gauss . . . 39

5 Élimination gaussienne **43**
- 5.1 Décomposition LU . . . 43
- 5.2 Analyse de l'erreur . . . 43
- 5.3 Pivotage . . . 43
- 5.4 Affinage de la solution . . . 45

6 Systèmes particuliers **49**
6.1 Décompositions LDM^T et LDL^T 49
6.2 Systèmes définis positifs 51
6.3 Systèmes symétriques indéfinis 55

7 Systèmes surdéterminés **57**
7.1 Le problème mathématique des moindres carrés 57
7.2 Orthogonalisations de Householder et de Gram–Schmidt 69
7.3 Orthogonalisation de Givens 76
7.4 Méthode QR avec pivotage sur les colonnes 76
7.5 Méthode de la décomposition en valeurs singulières 81
7.6 Étalonnage et équilibrage des systèmes et affinage des solutions . 81

8 Problème spectral des matrices nonsymétriques **83**
8.1 Théorie de la perturbation 83
8.2 Méthode de la puissance 83
8.3 Décomposition de Hessenberg et de Schur 83
8.4 Algorithme QR 83
8.5 Algorithme QZ et le problème spectral généralisé 83

9 Problème spectral des matrices symétriques **85**
9.1 Théorie des perturbations 85
9.2 Tridiagonalisation et algorithme QR symétrique 85
9.3 Méthode de Jacobi 85
9.4 Problème spectral généralisé 85

10 Méthodes de récurrence pour les systèmes linéaires **87**
10.1 Méthodes de Jacobi et de Gauss–Seidel 87
10.2 Méthode du gradient conjugué 87

11 Matrices tridiagonales **89**
11.1 Introduction 89
11.2 Tridiagonalisation d'une matrice symétrique 90
11.3 Décomposition LU 93
11.4 Valeurs propres 95
11.4.1 La méthode QR avec translation. 96
11.4.2 La suite de Sturm d'une matrice tridiagonale. 99
11.4.3 Le calcul des valeurs propres par la bisection. 103

Chapitre 1

Eléments d'algèbre des matrices

1.1 Vecteurs et matrices

Le *produit scalaire*: $x, y \in \mathbf{R}^n$,

$$x^T y := \sum_{i=1}^{n} x_i y_i = y^T x \in \mathbf{R}.$$

Le *produit extérieur*: $x \in \mathbf{R}^m, y \in \mathbf{R}^n$,

$$xy^T := \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} [y_1 \cdots y_n] = \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & \vdots & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{bmatrix} \in \mathbf{R}^{m \times n}.$$

1.2 Indépendance, orthogonalité, sous-espace

Le sous-espace engendré par $\{a_1, \ldots, a_n\}, a_j \in \mathbf{R}^m$, est l'*enveloppe linéaire*:

$$\text{lin}\{a_1, \ldots, a_n\} := \left\{ \sum_{j=1}^{n} \beta_j a_j | \quad \beta_1, \ldots, \beta_n \in \mathbf{R} \right\}.$$

Exemple 1.1 *Voici trois exemples importants de sous-espaces. Soit* $A \in \mathbf{R}^{m \times n}$.

(a) *L'*image *de A:*

$$\text{R}(A) := \{y \in \mathbf{R}^m | \quad y = Ax, \quad \text{un } x \in \mathbf{R}^n\}.$$

(b) *Le* noyau *de A:*

$$\text{N}(A) := \{x \in \text{R}^n | \quad Ax = 0\}.$$

(c) *Le* complémentaire orthogonal *d'un sous-espace,* $S \subset \mathbf{R}^m$*:*

$$S^{\perp} := \{y \in \mathbf{R}^m; \quad y^T x = 0, \forall x \in S\}.$$

Remarque 1.1 *Soit* $A = [a_1, \ldots, a_n]$*, alors*

$$\mathrm{R}(A) = \mathrm{lin}\{a_1, \ldots, a_n\},$$

c'est-à-dire Ax *est une combinaison linéaire des colonnes de* A*:*

$$Ax = x_1 a_1 + \cdots + x_n a_n.$$

Le rang *de* A *noté* $\mathrm{rg}(A)$ *est le nombre:*

$$\mathrm{rg}(A) := \dim[\mathrm{R}(A)].$$

Théorème 1.1 *Soit* $A \in \mathbf{R}^{m \times n}$*; alors*

$$\mathrm{rg}(A) + \dim[\mathrm{N}(A)] = n$$

et

$$\mathrm{R}(A)^{\perp} = \mathrm{N}(A^T).$$

1.3 Matrices spéciales

Soit $A \in \mathbf{R}^{m \times n}$; on dit que A est:

diagonale si $a_{ij} = 0, i \neq j$;

tridiagonale si $a_{ij} = 0, |i - j| > 1$;

triangulaire supérieure si $a_{ij} = 0, i > j$;

triangulaire supérieure stricte si $a_{ij} = 0, i \geq j$;

Hessenberg supérieure si $a_{ij} = 0, i > j + 1$.

Soit $A = (a_{ij})$ une matrice diagonale: $a_{ij} = 0, i \neq j$. On note $a_i := a_{ii}$ et

$$A := \mathrm{diag}(a_1, \ldots, a_k), \qquad k = \min\{m, n\}.$$

Soit $A \in \mathbf{R}^{n \times n}$; on dit que A est:

à *diagonale dominante* si

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \qquad \forall i;$$

une *matrice de permutation* si

$$A = [e_{s_1}, \ldots, e_{s_n}],$$

où $e_{s_i} = (0, \ldots, 1_{s_i}, 0, \ldots, 0)^T$ et $(s_1, \ldots, s_n)$ est une permutation de $(1, \ldots, n)$.

1.4 Matrices blocs

On note une *matrice bloc* $p \times q$

$$A = [A_{ij}] = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1q} \\ A_{21} & A_{22} & \cdots & A_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pq} \end{bmatrix},$$

où $A_{ij} \in \mathbf{R}^{m_i \times n_j}, \quad i = 1, \ldots, p, \quad j = 1, \ldots, q.$

Exemple 1.2

(a) Matrice bloc 2×2 triangulaire supérieure:

$$\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

(b) Matrice bloc 2×2 générale avec indication des dimensions des blocs:

$$\begin{array}{cc} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} & \begin{matrix} r \\ s \end{matrix} \\ \begin{matrix} u & v \end{matrix} & \end{array}$$

où A_{11}, par exemple, est un bloc $r \times u$.

Exercice 1.1

(a) Soit une matrice unitaire $Q = Q_1 + iQ_2, \quad Q_j \in \mathbf{R}^{n \times n}$; alors

$$Z = \begin{bmatrix} Q_1 & -Q_2 \\ Q_2 & Q_1 \end{bmatrix} \in \mathbf{R}^{2n \times 2n}$$

est orthogonale.

(b) Soit une matrice hermitienne $A = A^H := \bar{A}^T \in \mathbf{C}^{n \times n}$; alors A s'écrit sous la forme:

$$A = P + iQ,$$

où $P = P^T \in \mathbf{R}^{n \times n}$ est symétrique et $Q = -Q^T \in \mathbf{R}^{n \times n}$ est anti-symétrique.

(c) Soit $x, y \in \mathbf{R}^n$. Montrer que la matrice $xy^T \in \mathbf{R}^{n \times n}$ est de rang 1.

Chapitre 2

Normes et valeurs singulières

2.1 Normes de vecteurs

Définition 2.1 *Soient $x, y \in \mathbf{R}^n$ et $\alpha \in \mathbf{R}$. Une* norme *sur $\mathbf{R}^n$ satisfait les conditions qui suivent:*

$$\begin{aligned} \|x\| &\geq 0 \ \forall x \in \mathbf{R}^n, \qquad = 0 \quad ssi \quad x = 0, \\ \|x+y\| &\leq \|x\| + \|y\|, \\ \|\alpha x\| &= |\alpha| \, \|x\|. \end{aligned}$$

Pour p, $1 \leq p \leq \infty$, on a les normes:

$$\begin{aligned} \|x\|_p &= (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}, \qquad 1 \leq p < \infty, \\ \|x\|_\infty &= \max\{|x_1|, \ldots, |x_n|\}. \end{aligned}$$

En particulier, pour $p = 2$,

$$\|x\|_2 = \sqrt{|x_1|^2 + \cdots + |x_n|^2} = \sqrt{x^T x}.$$

On a les inégalités qui suivent.

(1) L'*inégalité de Hölder*: Soient $x, y \in \mathbf{R}^n$ et $1/p + 1/q = 1$, alors

$$\left|x^T y\right| \leq \|x\|_p \, \|y\|_q, \quad = \|x\|_p \, \|y\|_q \text{ ssi } x = \alpha y.$$

Remarque 2.1 La démonstration de l'inégalité de Hölder fait appel à des idées géométriques assez complexes.

(2) L'*inégalité de Cauchy-Schwarz* ($p = 2$):

$$\left|x^T y\right| \leq \|x\|_2 \, \|y\|_2, \quad = \|x\|_2 \, \|y\|_2 \text{ ssi } x = \alpha y.$$

Démonstration de l'inégalité de Cauchy-Schwarz:

$$\begin{aligned} 0 &\leq \|x+\lambda y\|_2^2 \\ &= \left(x^T+\lambda y^T\right)(x+\lambda y) \\ &= x^Tx+2\lambda x^Ty+\lambda^2 y^Ty, \qquad \left(\text{du fait que } x^Ty=y^Tx=\sum_{i=1}^n x_iy_i\right), \\ &=: p(\lambda); \end{aligned}$$

donc le discriminant est négatif:

$$4\left(x^Ty\right)^2-4\left(x^Tx\right)\left(y^Ty\right)\leq 0,$$

c'est-à-dire

$$\left|x^Ty\right|\leq \|x\|_2\,\|y\|_2.$$

Si $x=\alpha y$, l'égalité est évidente. □

Remarque 2.2 Soit $Q\in\mathbf{R}^{n\times n}$ orthogonale, alors

$$\|Qx\|_2=\|x\|_2.$$

On donne une démonstration géométrique et une démonstration algébrique de cette remarque

(1) **Démonstration géométrique**: Puisque l'action de Q sur x est une rotation ou une réflexion, Q ne change pas la norme de x.

(2) **Démonstration algébrique**: Puisque $Q^T=Q^{-1}$,

$$\|Qx\|_2=(Qx)^TQx=x^TQ^TQx=x^Tx=\|x\|_2. \quad \square$$

Remarque 2.3 Pour n fini, toutes les normes de $\mathbf{R}^n$ sont équivalentes, c'est-à-dire, pour les normes $\|\cdot\|_\alpha$ et $\|\cdot\|_\beta$ données, il existe deux constantes positives, c_1 et c_2, telles que

$$c_1\|x\|_\alpha\leq\|x\|_\beta\leq c_2\|x\|_\alpha.$$

En particulier pour les p-normes, $p=1,2,\infty$:

$$\begin{array}{llll} \|x\|_2 & \leq \|x\|_1 & \leq \sqrt{n}\,\|x\|_2, \\ \|x\|_\infty & \leq \|x\|_2 & \leq \sqrt{n}\,\|x\|_\infty, \\ \|x\|_\infty & \leq \|x\|_1 & \leq n\,\|x\|_\infty. \end{array}$$

On définit l'*erreur absolue* de la valeur approchée $\hat{x}\in\mathbf{R}^n$ par la différence:

$$\epsilon_a=\|\hat{x}-x\|$$

et l'*erreur relative* de $\hat{x}$ par le quotient:

$$\epsilon_r=\frac{\|\hat{x}-x\|}{\|x\|}, \qquad x\neq 0.$$

2.2 Normes de matrices

Soit $A \in \mathbf{R}^{m\times n}$. On a alors les normes p de A *subordonnées* aux normes p sur les vecteurs, $\|x\|_p$, $\quad 1 \le p \le \infty$,

$$\begin{aligned} \|A\|_p &= \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} \\ &= \sup_{\|x\|_p=1} \|Ax\|_p, \qquad \text{par l'homogénéité des normes.} \end{aligned}$$

La *trace* de $B \in \mathbf{R}^{m\times m}$, notée $\operatorname{tr} B$, est la somme des éléments de la diagonale de B:

$$\operatorname{tr} B = \sum_{i=1}^{m} b_{ii}.$$

On a aussi la norme nonsubordonnée de Frobenius, appelée aussi de Schur ou encore euclidienne:

$$\begin{aligned} \|A\|_F^2 &= \operatorname{tr} AA^T = \operatorname{tr} A^T A \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2. \end{aligned}$$

Remarque 2.4 Les normes p et F satisfont l'inégalité qui suit:

$$\|AB\| \le \|A\|\,\|B\|,$$

où $A \in \mathbf{R}^{m\times n}$ et $B \in \mathbf{R}^{n\times q}$.

On calcule facilement les normes $\|A\|_1$ et $\|A\|_\infty$:

$$\begin{aligned} \|A\|_1 &= \max_j \sum_{i=1}^{m} |a_{ij}|, \qquad \text{colonne maximum,} \\ \|A\|_\infty &= \max_i \sum_{j=1}^{n} |a_{ij}|, \qquad \text{ligne maximum.} \end{aligned}$$

En effet,

$$\|Ax\|_1 = \|x_1a_1 + \cdots + x_na_n\|_1 \text{ où } \|x\|_1 = |x_1| + \cdots + |x_n| = 1;$$

alors par l'inégalité du triangle:

$$\begin{aligned} \|Ax\|_1 &\le |x_1|\,\|a_1\|_1 + \cdots + |x_n|\,\|a_n\|_1 \\ &\le (|x_1| + \cdots + |x_n|) \max_j \|a_j\|_1 \\ &= \max_j \|a_j\|_1 \\ &= \|a_{j_0}\|_1, \qquad \text{pour un } j_0, \end{aligned}$$

et le maximum est atteint avec $x = e_{j_0}$.
De même,

$$\begin{aligned}
\|Ax\|_\infty &= \|[l_1^T x \; l_2^T x \; \cdots \; l_m^T x]^T\|_\infty \text{ où } \|x\|_\infty = \max_{1\le j\le n} |x_j| = 1, \\
&= \max\left\{\left|l_1^T x\right|, \left|l_2^T x\right|, \ldots, \left|l_m^T x\right|\right\} \\
&= \left|l_{i_0}^T x\right| \\
&\le \max_{0\le j\le n} |x_j| \, \|l_{i_0}\|_1 \\
&\le \|l_{i_0}\|_1, \qquad \text{puisque } |x_j| \le 1, \\
&\le \max_{1\le i\le n} \|l_i\|_1 = \|l_{i_1}\|_1 \\
&= \|A\|_\infty;
\end{aligned}$$

et le maximum est atteint avec $x = (\operatorname{sgn} a_{i_1,1}, \ldots, \operatorname{sgn} a_{i_1,n})^T$.

Remarque 2.5 Soient $A \in \mathbf{R}^{m\times n}$ quelconque et $U \in \mathbf{R}^{m\times m}$ et $V \in \mathbf{R}^{n\times n}$ orthogonales; alors

$$\|UAV\|_2 = \|A\|_2, \qquad \|UAV\|_F = \|A\|_F.$$

On voit facilement que

$$\|A\|_2 \le \|A\|_F \le \sqrt{n}\, \|A\|_2.$$

2.3 Décomposition en valeurs singulières

Soit $B \in \mathbf{R}^{n\times n}$ une matrice carrée; on note $\lambda(B)$ le *spectre* de B, c'est-à-dire l'ensemble des valeurs propres de B:

$$\lambda(B) = \{\lambda_1, \ldots, \lambda_n\}.$$

Considérons une matrice $A \in \mathbf{R}^{m\times n}$ et les matrices symétriques $AA^T \in \mathbf{R}^{m\times m}$ et $A^TA \in \mathbf{R}^{n\times n}$. Les valeurs propres de AA^T et de A^TA sont respectivement

$$\lambda\left(AA^T\right) = \{\sigma_1^2 \ge \cdots \ge \sigma_r^2 \ge 0_{r+1} = \cdots = 0_m\},$$

$$\lambda\left(A^TA\right) = \left\{\sigma_1^2 \ge \cdots \ge \sigma_r^2 \ge 0_{r+1} = \cdots = 0_n\right\}.$$

Définition 2.2 *Soit $A \in \mathbf{R}^{m\times n}$ et notons $p = \min\{m, n\}$. Les* valeurs singulières *de A sont les p nombres*

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_p \ge 0.$$

Théorème 2.1 *Soient $A \in \mathbf{R}^{m\times n}$ et $p = \min\{m, n\}$. Il existe deux matrices orthogonales $U = [u_1, \ldots, u_m] \in \mathbf{R}^{m\times m}$ et $V = [v_1, \ldots, v_n] \in \mathbf{R}^{n\times n}$ qui décomposent A en valeurs singulières:*

$$U^TAV = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) =: \Sigma$$

et

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_p \ge 0.$$

Démonstration. Soient les vecteurs unités $x \in \mathbf{R}^n$ et $y \in \mathbf{R}^m$, $\|x\|_2 = 1$, $\|y\|_2 = 1$, tels que

$$Ax = \sigma y, \quad \text{où} \quad \sigma = \|A\|_2.$$

On peut trouver deux matrices orthogonales

$$V = [x, V_1] \in \mathbf{R}^{n\times n}, \quad U = [y, U_1] \in \mathbf{R}^{m\times m}.$$

Alors la structure de $U^T AV$ est triangulaire bloc:

$$A_1 := U^T AV = \begin{bmatrix} \sigma & w^T \\ 0 & B \end{bmatrix} \begin{matrix} 1 \\ m-1 \end{matrix}$$

$$\begin{matrix} 1 & n-1 \end{matrix}$$

c'est-à-dire

$$AV = A[x, V_1] = [\sigma y, AV_1]$$

et

$$U^T AV = \begin{bmatrix} y^T \\ U_1^T \end{bmatrix} \begin{bmatrix} \sigma y & AV_1 \end{bmatrix} = \begin{bmatrix} \sigma & w^T \\ 0 & B \end{bmatrix}$$

puisque les lignes de U_1^T sont orthogonales à y.
On montre que $w = 0$. Puisque

$$A_1 \begin{bmatrix} \sigma \\ w \end{bmatrix} = \begin{bmatrix} \sigma & w^T \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma \\ w \end{bmatrix} = \begin{bmatrix} \sigma^2 + w^T w \\ Bw \end{bmatrix},$$

alors

$$\begin{aligned} \left\| A_1 \begin{bmatrix} \sigma \\ w \end{bmatrix} \right\|_2^2 &= \left(\sigma^2 + w^T w\right)^2 + \|Bw\|_2^2 \\ &\geq \left(\sigma^2 + w^T w\right)^2 \\ &= \left\| \begin{bmatrix} \sigma \\ w \end{bmatrix} \right\|_2^2 \left\| \begin{bmatrix} \sigma \\ w \end{bmatrix} \right\|_2^2. \end{aligned}$$

Il s'ensuit

$$\|A_1\|_2^2 \geq \sigma^2 + w^T w;$$

or

$$\sigma^2 = \|A\|_2^2 = \|A_1\|_2^2;$$

donc

$$w = 0.$$

On complète la démonstration par induction sous l'hypothèse que le théorème est vrai pour $B \in \mathbf{R}^{(m-1)\times(n-1)}$. □

Définition 2.3 *On note σ_i, u_i et v_i, respectivement, les* valeurs singulières*, les* vecteurs singuliers à gauche *et les* vecteurs singuliers à droite *de la matrice A.*

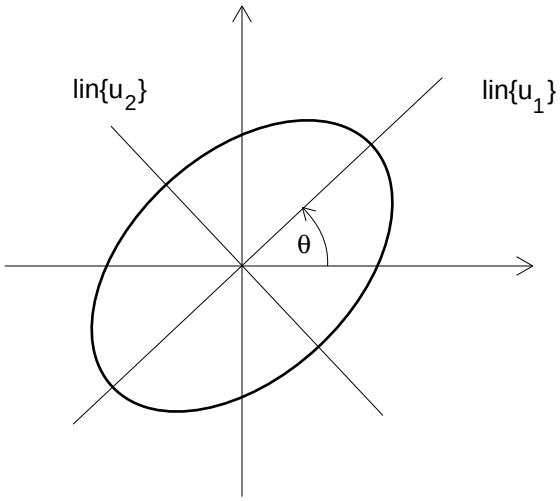

Figure 2.1: L'ellipsoïde E.

On voit que, pour $i = 1, 2, \ldots, p$,

$$Av_i = \sigma_i u_i,$$

$$A^T u_i = \sigma_i v_i, \quad \text{c'est-à-dire} \quad u_i^T A = \sigma_i v_i^T;$$

en fait,

$$U^T AV = \Sigma \Rightarrow AV = U\Sigma$$

et

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

Les vecteurs u_i et v_i sont les vecteurs propres respectivement de AA^T et $A^T A$.

Remarque 2.6 Les valeurs singulières de A sont les demi-longueurs des axes de l'ellipsoïde E:

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

Exemple 2.1 *Soit*

$$A = \begin{bmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{bmatrix}.$$

On a (v. fig. 2.1)

$$A = U\Sigma V^T = \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{bmatrix}^T,$$

où

$$\cos\theta = 0.8, \quad \sin\theta = 0.6, \quad \theta = \arctan 0.75 = 0.6435.$$

Corollaire 2.1 *Soit la décomposition en valeurs singulières de A, $U^T AV = \Sigma$. Si*

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

alors

(1) $rg\ A = r$,

(2) $N(A) = \text{lin}\{v_{r+1}, \ldots, v_n\}$,

(3) $R(A) = \text{lin}\{u_1, \ldots, u_r\}$,

(4) $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U_r \Sigma_r V_r^T$,

(5) $\|A\|_F^2 = \sigma_1^2 + \ldots + \sigma_r^2$,

(6) $\|A\|_2 = \sigma_1$.

Démonstration.

(1) U, V orthogonales $\Rightarrow$ rg $A =$ rg $U^T AV =$ rg $\Sigma = r$.

(2) $Av_k = U\Sigma V^T v_k = U\Sigma e_k = 0, \quad k = r+1, r+2, \ldots, n.$

(3) $A = U\Sigma V^T = [\sigma_1 u_1, \ldots, \sigma_r u_r, 0, \ldots, 0]V^T$;

alors $R(A) = \text{lin}(u_1, \ldots, u_r)$, puisque V^T est orthogonale.

(4) $$A = U\Sigma V^T = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 v_1^T \\ \cdots \\ \sigma_r v_r^T \\ 0^T \\ \cdots \\ 0^T \end{bmatrix} = \sigma_1 u_1 v_1^T + \cdots + \sigma_r u_r v_r^T.$$

Remarque 2.7 $u_i v_i^T$ est une projection P_i sur u_i, c'est-à-dire $u_i v_i^T x = \left(v_i^T x\right) u_i$ et $N(P_i) = \mathbf{R}^n \backslash \{v_i\}$.

(5) $\|A\|_F^2 = \|U\Sigma V^T\|_F^2 = \|\Sigma\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2$.

(6) $\|A\|_2 = \|U\Sigma V^T\|_2 = \|\Sigma\|_2 = \max_{\|x\|_2=1} \|\Sigma x\|_2 = \|\Sigma e_1\|_2 = \sigma_1$. □

2.4 Quantification de la presque déficience

On quantifie la presque déficience d'une matrice.

Corollaire 2.2 *Soit $U^T AV = \Sigma$. Si*

$$k < r = \text{rg } A$$

et

$$A_k := \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

alors

$$\min_{rg\ B=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

Démonstration. On a

$$U^T A_k V = \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0) \Rightarrow \operatorname{rg} A_k = k$$

et, avec $p = \min\{m, n\}$,

$$\|A - A_k\|_2 = \|U^T(A - A_k)V\|_2 = \|\operatorname{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_p)\|_2 = \sigma_{k+1}.$$

Si

$$\operatorname{rg} B = k, \qquad B \in \mathbf{R}^{m\times n},$$

alors, par un théorème du chapitre précédent: $\operatorname{rg} A + \dim N(A) = n$, il existe des vecteurs orthogonaux, $x_1, \ldots, x_{n-k}$, tels que

$$N(B) = \operatorname{lin}\{x_1, \ldots, x_{n-k}\}.$$

Le calcul des dimensions donne

$$\operatorname{lin}\{x_1, \ldots, x_{n-k}\} \cap \operatorname{lin}\{v_1, \ldots, v_{k+1}\} \neq \{0\}.$$

Donc cette envelope linéaire contient un vecteur unité, z. Or

$$Bz = 0$$

et

$$Az = \sum_{i=1}^{k+1} \sigma_i \left(v_i^T z\right) u_i;$$

donc

$$\|A - B\|_2^2 \geq \|(A - B)z\|_2^2 = \|Az\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 \left(v_i^T z\right)^2 \geq \sigma_{k+1}^2.$$

On a employé le fait que

$$\sum_{i=1}^{k+1} \left(v_i^T z\right)^2 = \|z\|^2 = 1.$$

2.5 Conditionnement des systèmes carrés

Soient $A \in \mathbf{R}^{n\times n}$ régulière et $b \in \mathbf{R}^n$. On considère le système linéaire $Ax = b$. Quel est l'effet des perturbations de A et de b sur la solution x?
Pour répondre à cette question on a recours à la décomposition de A en valeurs singulières:

$$A = U\Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v i^T.$$

Alors la solution s'exprime suivant la base $\{v_1, \ldots, v_n\}$:

$$x = A^{-1}b = V\Sigma^{-1}U^T b = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i.$$

Si σ_n est faible, une légère perturbation de A ou de b peut causer une grande perturbation de x.
En effet, notons

$$\cos\theta = \frac{\left|u_n^T b\right|}{\|b\|_2},$$

et perturbons le système $Ax = b$ dans la direction u_n:

$$\left(A - \epsilon u_n v_n^T\right) y = b + \epsilon \operatorname{sgn}\left(u_n^T b\right) u_n, \qquad \sigma_n > \epsilon \geq 0,$$

où

$$\operatorname{sgn} t = \begin{cases} +1, & \text{si } t > 0, \\ -1, & \text{si } t < 0, \end{cases}$$

désigne le signe de t. Alors on peut démontrer que l'erreur dans la solution satisfait:

$$\|y - x\|_2 \geq \frac{\epsilon}{\sigma_n} \|x\|_2 \cos\theta,$$

c'est-à-dire qu'une perturbation de l'ordre $O(\epsilon)$ peut entraîner une perturbation de l'ordre $O(\epsilon/\sigma_n)$ sur la solution.
La matrice du système perturbé s'écrit sous la forme:

$$\begin{aligned} A - \epsilon u_n v_n^T &= \sum_{i=1}^{n} \sigma_i u_i v_i^T - \epsilon u_n v_n^T \\ &= \sum_{i=1}^{n-1} \sigma_i u_i v_i^T + (\sigma_n - \epsilon) u_n v_n^T; \end{aligned}$$

on peut alors développer la solution y suivant la base $\{v_1, \ldots, v_n\}$:

$$y = \sum_{i=1}^{n-1} \frac{u_i^T b}{\sigma_i} v_i + \frac{u_n^T b + \epsilon \operatorname{sgn}\left(u_n^T b\right)}{\sigma_n - \epsilon} v_n.$$

Si l'on soustrait de y la solution x du système non perturbé exprimée suivant la base $\{v_1, \ldots, v_n\}$, on a l'erreur:

$$y - x = \left[\frac{u_n^T b + \epsilon \operatorname{sgn}\left(u_n^T b\right)}{\sigma_n - \epsilon} - \frac{u_n^T b}{\sigma_n} \right] v_n.$$

Il n'y a ici aucune surprise parce que σ_n est la distance de A à l'ensemble des matrices singulières; si A s'approche de cet ensemble, la solution devient instable.

Passons maintenant à une *mesure de l'instabilité* des systèmes. Considérons le système à un paramètre

$$(A + \epsilon F) x(\epsilon) = b + \epsilon f,$$

où $A, F \in \mathbf{R}^{n\times n}$ et $f \in \mathbf{R}^n$, et notons $x = x(0)$ la solution de $Ax = b$. Si A est régulière, alors $x(\epsilon) \in C^1$ dans un ouvert de 0. Un développement de Taylor de $x(\epsilon)$ d'ordre $O(\epsilon^2)$ donne:

$$Ax + \epsilon A\dot{x}(0) + \epsilon Fx = b + \epsilon f + O\left(\epsilon^2\right),$$

c'est-à-dire

$$\dot{x}(0) = A^{-1}(f - Fx) + O(\epsilon);$$

de plus

$$x(\epsilon) = x + \epsilon\dot{x}(0) + O\left(\epsilon^2\right);$$

donc

$$\begin{aligned}
x(\epsilon) - x &= \epsilon\dot{x}(0) + O\left(\epsilon^2\right) \\
&= \epsilon A^{-1}(f - Fx) + O\left(\epsilon^2\right), \\
\|x(\epsilon) - x\| &\leq \epsilon\|A^{-1}\|\{\|f\| + \|F\|\,\|x\|\} + O\left(\epsilon^2\right), \\
\frac{\|x(\epsilon) - x\|}{\|x\|} &\leq \epsilon\|A^{-1}\|\left\{\frac{\|f\|}{\|x\|} + \|F\|\right\} + O\left(\epsilon^2\right) \\
&\leq \epsilon\|A^{-1}\|\,\|A\|\left\{\frac{\|f\|}{\|b\|} + \frac{\|F\|}{\|A\|}\right\} + O\left(\epsilon^2\right),
\end{aligned}$$

puisque

$$Ax = b \quad \Rightarrow \quad \|A\|\,\|x\| \geq \|b\| \quad \Rightarrow \quad \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}.$$

Définition 2.4 *Le* conditionnement *de A dans la norme p est le nombre*

$$\kappa_p(A) = \mathrm{cond}_p(A) := \|A\|_p\,\|A^{-1}\|_p.$$

Notation 2.1 On note *erreur relative* en A, resp. en b, les nombres

$$\rho_A = \epsilon\,\frac{\|F\|}{\|A\|}, \quad \text{resp.} \quad \rho_b = \epsilon\,\frac{\|f\|}{\|b\|}.$$

Alors la borne supérieure de l'erreur relative en x qu'on vient d'obtenir s'écrit de la façon suivante:

$$\frac{\|x(\epsilon) - x\|}{\|x\|} = \kappa(A)[\rho_b + \rho_A] + O\left(\epsilon^2\right).$$

Exemple 2.2 *Soit*

$$A = \begin{bmatrix} 1.00 & 0.99 \\ 0.99 & 0.98 \end{bmatrix}, \quad b = \begin{bmatrix} 1.99 \\ 1.97 \end{bmatrix}, \quad \delta b = \begin{bmatrix} -0.000\,097 \\ 0.000\,106 \end{bmatrix}.$$

Alors la solution de $Ax = b$, resp. de $A\hat{x} = b + \delta b$, et δx sont:

$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} 3.000\,0 \\ -1.020\,3 \end{bmatrix}, \quad \delta x = \begin{bmatrix} 2.000\,0 \\ -2.020\,3 \end{bmatrix}.$$

On voit que

$$\|\delta b\|_2 \approx 10^{-4}\sqrt{2}, \qquad \frac{\|\delta b\|_2}{\|b\|_2} \approx \frac{10^{-4}}{2},$$

$$\frac{\|\delta x\|_2}{\|x\|_2} \approx 40\,000\,\frac{\|\delta b\|_2}{\|b\|_2} \approx 2.$$

La forte perturbation de x causée par une très faible perturbation de b s'explique par le très mauvais conditionnement de A:

$$\mathrm{cond}_2(A) = 39\,600 = \frac{\sigma_1}{\sigma_2} = \frac{1.980\,5}{0.000\,05} = \|A\|_2\|A^{-1}\|_2.$$

Théorème 2.2 *Soient $A \in \mathbf{R}^{n\times n}$ régulière et $b \in \mathbf{R}^n$. Considérons une perturbation $\Delta A \in \mathbf{R}^{n\times n}$ de A telle que*

$$\|\Delta A\|\,\|A^{-1}\| = r < 1.$$

Alors

$$(A + \Delta A)y = b + \Delta b$$

est un système régulier. De plus, s'il existe un $\delta \geq 0$ tel que

$$\|\Delta A\| \leq \delta\|A\|$$

et

$$\|\Delta b\| \leq \delta\|b\|,$$

alors

$$\frac{\|x - y\|}{\|x\|} \leq \frac{2\delta\kappa(A)}{1 - r}, \qquad x \neq 0.$$

Démonstration. Employer le lemme de Banach:

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

et l'inégalité, pour $\|E\| < 1$,

$$\|(I - E)^{-1}\| \leq \frac{1}{1 - \|E\|}.$$

Exercice 2.1

(a) Soit $A \in \mathbf{R}^{m\times n}$. Montrer:

$$\sigma_{\max} = \max_{y\in\mathbf{R}^m, x\in\mathbf{R}^n} \frac{y^T Ax}{\|y\|_2\|x\|_2}.$$

(b) Soit

$$A = \begin{bmatrix} w & x \\ y & z \end{bmatrix}.$$

Exprimer $\sigma_{\max}$ et $\sigma_{\min}$ en fonctions de w, x, y, z.

(c) Montrer que $A \in \mathbf{R}^{m\times n}$ quelconque est limite d'une suite de matrices $\{A_k\}$ de rang plein, c'est-à-dire $\operatorname{rg} A_k = \min\{m, n\}$.

(d) Soit $A \in \mathbf{R}^{m\times n}$ de rang n. Montrer:

$$\left\| A \left(A^T A\right)^{-1} A^T \right\|_2 = 1.$$

(e) Trouver la matrice de rang 1 la plus près de

$$A = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}$$

dans la norme de Frobenius.

(f) Soit $A \in \mathbf{R}^{m\times n}$. Montrer:

$$\|A\|_F \leq \sqrt{\operatorname{rg} A}\, \|A\|_2.$$

Remarque: Cette inégalité est plus forte que l'inégalité suivante:

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\, \|A\|_2.$$

Chapitre 3

Calculs en virgule flottante

3.1 Nombres en virgule flottante

On emploie la terminologie qui suit:
la *base* β, un entier > 1;
la *précision* t égale au nombre de chiffres significatifs (v. sect. suiv.);
la *puissance minimum* L de β;
la *puissance maximum* U de β.

Notation 3.1 *On note* $\mathcal{F}$ *l'*ensemble des nombres en virgule flottante*; alors tout* $f \in \mathcal{F}$ *s'écrit d'une et d'une seule façon:*

$$f = \begin{cases} \pm.d_1 d_2 \cdots d_t \times \beta^e, & 0 \leq d_i < \beta, d_1 \neq 0, L \leq e \leq U, \\ 0. \end{cases} \tag{3.1}$$

Lemme 3.1 Tout $f \in \mathcal{F} - \{0\}$ satisfait les inégalités:

$$\beta^{L-1} = m \leq |f| \leq M = \beta^U \left(1 - \beta^{-t}\right), \tag{3.2}$$

où m désigne le plus petit nombre > 0 et M le plus grand nombre respectivement de $\mathcal{F}$.

Démonstration. On détermine m et M.

$$m = 0.100 \cdots 0_t \times \beta^L = \beta^{-1}\beta^L = \beta^{L-1};$$

$$\begin{aligned} M &= 0.(\beta-1)_1 \cdots (\beta-1)_t \times \beta^U \\ &= [(\beta-1)\beta^{-1} + (\beta-1)\beta^{-2} + \cdots + (\beta-1)\beta^{-t}] \times \beta^U \\ &= [1 - \beta^{-1} + \beta^{-1} - \beta^{-2} + \cdots + \beta^{-(t-1)} - \beta^{-t}] \times \beta^U \\ &= \beta^U(1 - \beta^{-t}), \end{aligned}$$

puisque la somme se syncope. □

Figure 3.1: Les nombres positifs et zéro de l'univers des nombres caractérisés par $\beta = 2$, $t = 3$, $L = -1$, $U = 2$.

Exemple 3.1 *Soient* $\beta = 10, t = 4, L = -5, U = 6$*; alors*

$$f = -.8765 \times 10^2 \in \mathcal{F}.$$

On a $m = 10^{-5-1} = 10^{-6}$ *et* $M = 10^6\left(1 - 10^{-4}\right) = 0.9999 \times 10^6$.

Exemple 3.2 *Soient* $\beta = 2, t = 3, L = -1$ *et* $U = 2$. *Enumérer tous les éléments* $\mathcal{F}$.

Résolution.

$$m = 2^{-1-1} = 2^{-2} = \frac{1}{4}, \qquad M = 2^2\left(1 - 2^{-3}\right) = 4\left(1 - \frac{1}{8}\right) = \frac{7}{2}.$$

Voici tous les nombres positifs non nuls de $\mathcal{F}$:

$$\begin{array}{lcl}
0.100 \times 2^{-1} & = & 4/16 = m \\
0.101 \times 2^{-1} & = & 5/16 \\
0.110 \times 2^{-1} & = & 6/16 \\
0.111 \times 2^{-1} & = & 7/16 \\
0.100 \times 2^{0} & = & 8/16 = 4/8 \\
0.101 \times 2^{0} & = & 5/8 \\
0.110 \times 2^{0} & = & 6/8 \\
0.111 \times 2^{0} & = & 7/8 \\
0.100 \times 2^{1} & = & 8/8 = 4/4 \\
0.101 \times 2^{1} & = & 5/4 \\
0.110 \times 2^{1} & = & 6/4 \\
0.111 \times 2^{1} & = & 7/4 \\
0.100 \times 2^{2} & = & 8/4 = 4/2 \\
0.101 \times 2^{2} & = & 5/2 \\
0.110 \times 2^{2} & = & 6/2 \\
0.111 \times 2^{2} & = & 7/2 = M.
\end{array}$$

L'ensemble $\mathcal{F}$ contient donc 16 nombres positifs, 16 nombres négatifs et zéro: en tout 33 nombres inégalement espacés (v. fig. 3.1).

Exercice 3.1 Soit $\beta = 3$, $t = 2$, $L = -1$ et $U = +1$.

(a) Dresser la liste des nombres positifs de l'univers $\mathcal{F}$.

(b) Calculer la valeur de m et de M

3.2 Ordinateur modèle

Soit le sous-ensemble des nombres de la droite réelle:

$$\mathcal{G} = \{x \in \mathbf{R} \mid m \leq |x| \leq M\} \cup \{0\} \tag{3.3}$$

et l'application

$$\mathrm{fl} : \mathcal{G} \to \mathcal{F} \tag{3.4}$$

définie par

$$\mathrm{fl}(x) = \begin{cases} x_a \in \mathcal{F} \text{ le plus près de } x \text{ (en s'éloignant de l'origine en cas d'égalité),} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{si l'on arrondit}, \\ x_t \in \mathcal{F} \text{ le plus près de } x \text{ tel que } |x_t| \leq |x|, \ \textbf{si l'on tronque}. \end{cases}$$

Théorème 3.1 *Soit $x \in \mathcal{G}$ et* **u** *l'unité d'arrondi, alors*

$$fl(x) = x(1+\epsilon), \qquad |\epsilon| \leq \mathbf{u}, \tag{3.5}$$

où

$$\mathbf{u} = \begin{cases} \frac{1}{2}\beta^{1-t}, & \textit{si l'on arrondit}, \\ \beta^{1-t}, & \textit{si l'on tronque}. \end{cases} \tag{3.6}$$

Démonstration.
(a) On considère d'abord le cas où l'on arrondit. Le cas où $x = 0$ est trivial. Si $x < 0$, on prend $-x > 0$. Donc on se restreint aux valeurs de x satisfaisant l'inégalité:

$$\beta^{L-1} \leq x < \beta^U.$$

Soit e l'entier, unique, tel que

$$\beta^{e-1} \leq x < \beta^e.$$

On détermine l'incrément entre deux nombres consécutifs de $\mathcal{F}$ dans l'intervalle

$$\left[\beta^{e-1}, \beta^e\right[.$$

Puisque

$$\beta^{e-1} = 0.10\cdots 0_t \times \beta^e,$$

on obtient le nombre suivant en ajoutant à β^{e-1} l'incrément

$$0.0_1 0_2 \cdots 0_{t-1} 1_t \times \beta^e = \beta^{e-t}.$$

Or cet incrément est uniforme sur tout cet intervalle puisque la puissance e est fixée. Donc, sur cet intervalle l'incrément est β^{e-t}. Il suit que

$$|\,\mathrm{fl}(x) - x| \leq \frac{1}{2}\beta^{e-t}. \tag{3.7}$$

Puisque

$$\beta^{e-1} \leq x,$$

on obtient

$$\frac{|\operatorname{fl}(x) - x|}{|x|} \leq \frac{\frac{1}{2}\beta^{e-t}}{\beta^{e-1}} = \frac{1}{2}\beta^{1-t};$$

et puisque

$$\epsilon = \frac{\operatorname{fl}(x) - x}{x},$$

on a démontré le théorème si l'on arrondit.
(b) Si l'on tronque, (3.7) devient

$$|\operatorname{fl}(x) - x| \leq \beta^{e-t}. \quad \square$$

Corollaire 3.1 *Il y a $\beta^t - \beta^{t-1}$ nombres de $\mathcal{F}$ dans l'intervalle*

$$\left[\beta^{e-1}, \beta^e\right[.$$

Démonstration. Puisque l'incrément entre deux nombres de $\mathcal{F}$ dans cet intervalle est β^{e-t}, alors

$$\frac{\beta^e - \beta^{e-1}}{\beta^{e-t}} = \beta^t - \beta^{t-1}. \qquad \square$$

Exemple 3.3 *Soient $\beta = 10$, $t = 4$, $x = \pi$ et l'on arrondit. Vérifier* (3.7) *pour*

$$x = 0.314\,159 \times 10^1$$

Résolution. On voit que

$$x = 0.314\,159 \times 10^1 \Rightarrow e = 1.$$

Donc

$$\operatorname{fl}(x) = 0.3142 \times 10^1,$$

$$\epsilon = \frac{\operatorname{fl}(x) - x}{x} = 0.000\,130\,507\ldots.$$

Or

$$\frac{1}{2}\beta^{1-t} = \frac{1}{2} \times 10^{1-4} = 0.5 \times 10^{-3} = 0.000\,5;$$

alors

$$|\epsilon| \leq \frac{1}{2}\beta^{1-t}. \quad \square$$

Appliquons aux opérations arithmétiques sur les nombres de $\mathcal{F}$ le théorème qu'on vient d'établir sur l'erreur relative de l'opération

$$\mathcal{G} \ni x \mapsto \operatorname{fl}(x) \in \mathcal{F}.$$

Notation 3.2 *Notons* op *les opérations exactes* $+$, $-$, $\times$ *et* $\div$ *qui sont sur des registres et des accumulateurs très longs et donc qui ne causent aucune erreur de chute.*

Soient $a, b \in \mathcal{F}$; on considère $a \,\mathrm{op}\, b$. Il y a deux cas:

(i) si $a \,\mathrm{op}\, b \notin \mathcal{G}$, on a un *dépassement de capacité*
supérieur si $|a \,\mathrm{op}\, b| > M$ (ceci produit souvent un arrêt du programme),
inférieur si $0 < |a \,\mathrm{op}\, b| < m$ (dans ce cas beaucoup d'ordinateurs réduisent le résultat à zéro);

(ii) si $a \,\mathrm{op}\, b \in \mathcal{G}$, alors

$$\mathrm{fl} : a \,\mathrm{op}\, b \mapsto \mathrm{fl}(a \,\mathrm{op}\, b) \in \mathcal{F}.$$

Corollaire 3.2 *Si* $a, b \in \mathcal{F}$ *et* $a \,\mathrm{op}\, b \in \mathcal{G}$, *alors*

$$fl(a \,\mathrm{op}\, b) = (a \,\mathrm{op}\, b)(1 + \epsilon), \qquad |\epsilon| \leq \mathbf{u}; \tag{3.8}$$

donc l'erreur relative d'une seule opération est petite:

$$|\epsilon| = \frac{|\, fl(a \,\mathrm{op}\, b) - (a \,\mathrm{op}\, b)|}{|a \,\mathrm{op}\, b|} \leq \mathbf{u}, \qquad a \,\mathrm{op}\, b \neq 0.$$

Exemple 3.4 *Soient* $\beta = 10$ *et* $t = 3$*; si l'on tronque,*

$$fl[fl(10^{-3} + 1) - 1] = fl[0.100 \times 10^{1} - 1] = fl[0] = 0;$$

et

$$fl[10^{-3} + fl(1 - 1)] = 10^{-3}.$$

Dans le premier cas, l'erreur relative est 100% et dans le second 0%. Donc les opérations en virgule flottante ne sont pas toujours associatives.

3.3 Nombre de chiffres significatifs

Définition 3.1 *Le* nombre de chiffres significatifs *en virgule flottante, c'est le nombre de chiffres comptés du premier chiffre non nul au dernier chiffre non nul.*

Exemple 3.5 *Si* $\beta = 10$ *et* $t = 5$, *alors:*

$$+0.12345 \times 10^{7} \quad \textit{a 5 chiffres significatifs}$$

et

$$-0.12000 \times 10^{-3} \quad \textit{a 2 chiffres significatifs.}$$

Deux situations de perte de chiffres significatifs peuvent se produire: la perte catastrophique du nombre de chiffres significatifs dans la soustraction de deux nombres presqu'égaux et le résultat illusoire de la division.

Exemple 3.6 *Soient $\beta = 10$ et $t = 5$:*

$$+0.12345 \times 10^5 - \left(+0.12342 \times 10^5\right) = +0.30000 \times 10^1.$$

Il y a une perte de 4 chiffres significatifs.
L'exemple suivant illustre d'une façon dramatique la perte du nombre de chiffres significatifs du fait que les termes d'une somme grands en module et de signes opposés s'annulent tour à tour.

Exemple 3.7 *Soient $\beta = 10$ et $t = 5$; évaluer $e^{-5.5}$ au moyen de la série de Taylor:*

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!}$$

en $x = 5.5$.

Résolution. Puisque $e^{-5.5} \approx 0.004\,086\,771\,4$ à 8 chiffres significatifs près et

$$\frac{(-5.5)^{26}}{26!} = 0.000\,000\,044\,0,$$

il suffit de prendre $N = 25$. On obtient alors, en virgule flottante:

$$\begin{aligned} e^{-5.5} \quad \approx \quad & + \ 1.000\,0 \\ & - \ 5.500\,0 \\ & +15.125 \\ & -27.730 \\ & +38.129 \\ & -41.942 \\ & +38.446 \\ & -30.208 \\ & +20.768 \\ & -12.692 \\ & + \ 6.980\,3 \\ & - \ 3.490\,2 \\ & + \ 1.599\,7 \\ & \quad \vdots \\ \approx \quad & + \ 0.002\,636\,3 \end{aligned}$$

Donc la réponse n'a aucun chiffre correct. De fait, l'erreur d'arrondi dans les gros termes, tel 38.129, est aussi grande que la réponse elle-même. Plutôt que

d'augmenter la valeur de t à 10 et même 11, on peut, pour éviter la perte de chiffres significatifs, employer l'algorithme bien conçu suivant:

$$e^{-5.5} = \frac{1}{e^{5.5}} = \frac{1}{1 + 5.5 + \cdots} \approx 0.004\,086\,5;$$

dans ce cas l'erreur relative n'est que de 0.007% avec $t = 5$.

Voici un quotient illusoire. Soient $\beta = 10$, $t = 5$ et $L = -15$. Alors:

$$a = +0.12345 \times 10^{-15} \approx 0, \quad b = +0.12345 \times 10^{-14} \approx 0;$$

mais

$$\frac{a}{b} = +0.10000 \times 10^{0} \approx \frac{0}{0}.$$

3.4 Effet cumulatif des erreurs

Dans ce paragraphe, le produit scalaire servira d'exemple de l'accumulation des erreurs dans une suite d'opérations en virgule flottante.

Définition 3.2 *Soient $x, y \in \mathbf{R}^n$. L'opération en virgule flottante:*

$$s := s + x_k y_k \tag{3.9}$$

sera appelée flop.

L'algorithme suivant calcule $x^T y$ en n flops:

Algorithme 3.1 *L'algorithme du produit scalaire $x^T y$:*

$$s := 0$$

$$\textit{Pour } \; k = 1, 2, \ldots, n$$

$$s := s + x_k y_k$$

Bornons maintenant l'erreur

$$\left| \mathrm{fl}\left(x^T y\right) - x^T y \right|.$$

Posons

$$s_p = \mathrm{fl}\left[\sum_{k=1}^{p} x_k y_k\right];$$

alors

$$s_1 = x_1 y_1 (1 + \delta_1), \qquad |\delta_1| \leq \mathbf{u},$$

et pour $p = 2, \ldots, n$,

$$\begin{aligned} s_p &= \mathrm{fl}[s_{p-1} + \mathrm{fl}(x_p y_p)] \\ &= [s_{p-1} + x_p y_p (1 + \delta_p)](1 + \epsilon_p), \qquad |\delta_p|, |\epsilon_p| \leq \mathbf{u}. \end{aligned}$$

Donc

$$\mathrm{fl}\left(x^T y\right) = s_n = \sum_{k=1}^{n} x_k y_k (1+\gamma_k)$$

où

$$1+\gamma_k = (1+\delta_k)\prod_{j=k}^{n}(1+\epsilon_j), \qquad \epsilon_1 = 0.$$

On écrit au long l'expression précédente:

$$\begin{aligned}
\mathrm{fl}\left(x^T y\right) &= \Big[\cdots\big\{\big[\big(x_1y_1(1+\delta_1)+x_2y_2(1+\delta_2)\big)(1+\epsilon_2)\\
&\qquad + x_3y_3(1+\delta_3)\big](1+\epsilon_3)\\
&\qquad + x_4y_4(1+\delta_4)\big\}(1+\epsilon_4)\\
&\qquad \vdots\\
&\qquad + x_ny_n(1+\delta_n)\Big](1+\epsilon_n)\\
&= \quad x_1y_1(1+\delta_1)(1+\epsilon_2)(1+\epsilon_3)\cdots(1+\epsilon_n)\\
&\quad + x_2y_2(1+\delta_2)(1+\epsilon_2)(1+\epsilon_3)\cdots(1+\epsilon_n)\\
&\quad + x_3y_3(1+\delta_3)\qquad (1+\epsilon_3)\cdots(1+\epsilon_n)\\
&\quad \vdots\\
&\quad + x_ny_n(1+\delta_n)\qquad\qquad (1+\epsilon_n).
\end{aligned}$$

Alors

$$\begin{aligned}
\mathrm{fl}\left(x^T y\right) &= \sum_{k=1}^{n} x_k y_k + \sum_{k=1}^{n} x_k y_k \gamma_k\\
&= x^T y + \sum_{k=1}^{n} x_k y_k \gamma_k
\end{aligned}$$

et

$$\begin{aligned}
\left|\mathrm{fl}\left(x^T y\right) - x^T y\right| &= \left|\sum_{k=1}^{n} x_k y_k \gamma_k\right|\\
&\le \sum_{k=1}^{n} |x_k|\,|y_k|\,|\gamma_k|\\
&\le n\mathbf{u}|x|^T\,|y| + O\left(\mathbf{u}^2\right)
\end{aligned}$$

du fait que

$$|\gamma_k| \le n\mathbf{u} + O\left(\mathbf{u}^2\right).$$

On a noté la *valeur absolue* $|x|$ du vecteur x:

$$|x| = \big(|x_1|, \ldots, |x_n|\big)^T.$$

Au moyen d'une analyse plus fine de l'erreur sous l'hypothèse additionnelle: $n\mathbf{u} \leq 0.01$, qui est très acceptable en pratique, on obtient la borne suivante:

$$\| \operatorname{fl}(x^T y) - x^T y\| \leq 1.01 n\mathbf{u}|x|^T|y|,$$

sans terme $O\left(\mathbf{u}^2\right)$.

Remarque 3.1 Si

$$\left|x^T y\right| \ll |x|^T\ |y|,$$

l'erreur relative dans $\operatorname{fl}\left(x^T y\right)$ n'est pas nécessairement petite, comme on peut voir à l'exemple suivant. Il est donc important d'*accumuler* les produits scalaires, c'est-à-dire d'employer la précision double, $\mathbf{u} = \beta^{1-2t}$, et d'arrondir ou tronquer le résultat final à t chiffres. Puisque le produit de deux nombres à t chiffres est un nombre à $2t$ chiffres, l'erreur est alors très faible. On peut montrer que si $n\mathbf{u}$ est suffisamment petit (disons $n\mathbf{u} \leq 0.01$) et si l'on accumule le produit scalaire, alors

$$\left|\operatorname{fl}_2\left(\sum_{i=1}^{n} x_i y_i\right) - \sum_{i=1}^{n} x_i y_i\right| \leq c\mathbf{u}\|x\|_2\,\|y\|_2, \tag{3.10}$$

où fl_2 désigne l'opération (3.4) en précision double, c'est-à-dire avec des mots de longueur $2t$ si $n\mathbf{u}$ est suffisamment petit.

Exemple 3.8 *On montre que si* $\left|x^T y\right| \ll |x|^T\ |y|$, *l'erreur relative dans* $fl_2\left(x^T y\right)$ *n'est pas nécessairement petite à cause des pertes importantes de chiffres significatifs. Soit* $\beta = 10$, $t = 4$ *et les vecteurs:*

$$x = \begin{bmatrix} 0.1002 \times 10^{-1} \\ 0.9999 \times 10^{+0} \\ 0.2000 \times 10^{-1} \end{bmatrix}, \qquad y = \begin{bmatrix} 0.1003 \times 10^{+1} \\ -0.9995 \times 10^{+0} \\ -0.2803 \times 10^{+1} \end{bmatrix}.$$

On accumule le produit scalaire avec $t = 8$,

$$\begin{aligned} fl_2(x^T y) &= x_1 y_1 + x_2 y_2 + x_3 y_3 \\ &= 0.1005\,0060 \times 10^{+1} - 0.0999\,4000\,5 \times 10^{+1} - 0.5606\,0000 \times 10^{-2} \\ &\qquad (\textit{on arrondit } x_2 y_2) \\ &= (0.1005\,0060 \times 10^{+1} - 0.0999\,4001\ \times 10^{+1}) - 0.5606\,0000 \times 10^{-2} \\ &= -0.1000 \times 10^{-6}. \end{aligned}$$

Le résultat, arrondi à $t = 4$, *est assez petit. Maintenant, on calcule* $|x|^T|y|$ *avec* $t = 4$ *parce qu'il n'y a pas de soustraction:*

$$\begin{aligned} |x|^T|y| &= 0.1005 \times 10^{+1} + 0.0999\,4 \times 10^{+1} + 0.0005\,606 \times 10^{+1} \\ &\qquad (\textit{on arrondit } x_2 y_2 \textit{ et } x_3 y_3) \\ &= (0.1005 \times 10^{+1} + 0.0999 \times 10^{+1}) + 0.0006 \times 10^{+1} \\ &= 0.2010 \times 10^{+1}. \end{aligned}$$

Le résultat est grand. La valeur exacte de $x^T y$:

$$x^T y = -0.5000 \times 10^{-7},$$

est petite. On calcule la différence:

$$\begin{aligned} fl_2(x^T y) - x^T y = &= -0.1000 \times 10^{-6} + 0.0500 \times 10^{-6} \\ &= -0.0500 \times 10^{-6} \\ &= -0.5000 \times 10^{-7}. \end{aligned}$$

Enfin l'erreur relative est grande:

$$\frac{|fl_2(x^T y) - x^T y|}{|x^T y|} = \frac{0.5 \times 10^{-7}}{0.5 \times 10^{-7}} = 100\%.$$

Il n'y a donc aucun chiffre correct dans le produit scalaire accumulé à cause des pertes importantes de chiffres significatifs.

On termine par une citation libre de Wilkinson: l'analyse de l'erreur a pour but premier de détecter l'instabillité des calculs plutôt que de borner l'erreur; les bornes théoriques sont souvent très pessimistes; les bornes pratiques sont *a posteriori* et tiennent compte de la distribution statistique des erreurs.

3.5 Erreurs d'arrondi des calculs matriciels

Notation 3.3 *Soit une matrice $E \in \mathbf{R}^{m \times n}$. On note:*

$$|E| := (|e_{ij}|) \tag{3.11}$$

la matrice dont les composantes sont les modules des composantes de E. On note alors l'inégalité:

$$|E| \leq |F| \Longleftrightarrow |e_{ij}| \leq |f_{ij}| \qquad \forall i, j. \tag{3.12}$$

Soient

$$A, B \in \mathcal{F}(\mathbf{R}^{m \times n}),$$

c'est-à-dire

$$a_{ij}, b_{ij} \in \mathcal{F}$$

et

$$\alpha \in \mathcal{F}.$$

Alors

$$\mathrm{fl}(\alpha A) = \alpha A + E, \qquad |E| \leq \mathbf{u}|\alpha A|,$$

$$\mathrm{fl}(A + B) = (A + B) + E, \qquad |E| \leq \mathbf{u}|A + B|.$$

Soient maintenant $A \in \mathcal{F}(\mathbf{R}^{m\times n})$ et $B \in \mathcal{F}(\mathbf{R}^{n\times p})$, c'est-à-dire les éléments de A et de B sont dans $\mathcal{F}$; alors

$$\mathrm{fl}(AB) = AB + E, \qquad |E| \leq n\mathbf{u}|A|\,|B| + O\left(\mathbf{u}^2\right). \tag{3.13}$$

En effet, si

$$(1+\alpha) = \prod_{k=1}^{n}(1+\alpha_k) = 1 + \sum_{k=1}^{n}\alpha_k + O\left(\sum \alpha_k^2\right)$$

et

$$|\alpha_k| \leq \mathbf{u},$$

alors

$$|\alpha| \leq n\mathbf{u} + O\left(\mathbf{u}^2\right)$$

et

$$\left|\mathrm{fl}\left(x^T y\right) - x^T y\right| \leq n\mathbf{u}|x|^T|y| + O\left(\mathbf{u}^2\right). \tag{3.14}$$

Remarque 3.2 L'erreur relative dans $\mathrm{fl}(AB)$ peut être grande si $|A|\,|B| \gg |AB|$. Donc il est important d'*accumuler* les produits scalaires lors du produit AB.

Remarque 3.3 (3.14) est un exemple d'analyse directe de l'erreur: on cherche une borne supérieure de la solution. Au paragraphe suivant on procède a posteriori.

3.6 Analyse de l'erreur rétrograde

L'analyse rétrograde de l'erreur, par opposition à l'analyse directe, rejette l'erreur sur les données. On suppose que la solution numérique est la solution exacte d'un problème voisin du problème original.

Exemple 3.9 *Soient*

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ 0 & b_{22} \end{bmatrix}.$$

Alors

$$fl(AB) = \begin{bmatrix} a_{11}b_{11}(1+\epsilon_1) & [a_{11}b_{12}(1+\epsilon_2) + a_{12}b_{22}(1+\epsilon_3)](1+\epsilon_4) \\ 0 & a_{22}b_{22}(1+\epsilon_5) \end{bmatrix},$$

où $|\epsilon_i| \leq \mathbf{u},\ i = 1,\ldots,5$.

Si l'on considère les matrices quelque peu perturbées:

$$\widehat{A} = \begin{bmatrix} a_{11} & a_{12}(1+\epsilon_3)(1+\epsilon_4) \\ 0 & a_{22}(1+\epsilon_5) \end{bmatrix}, \qquad \widehat{B} = \begin{bmatrix} b_{11}(1+\epsilon_1) & b_{12}(1+\epsilon_2)(1+\epsilon_4) \\ 0 & b_{22} \end{bmatrix}$$

on voit que

$$fl(AB) = \widehat{A}\widehat{B},$$

où

$$\widehat{A} = A + E, \qquad |E| \leq 2\mathbf{u}|A| + O\left(\mathbf{u}^2\right),$$

et

$$\widehat{B} = B + F, \qquad |F| \leq 2\mathbf{u}|B| + O\left(\mathbf{u}^2\right).$$

On peut dire que le produit sur ordinateur est le produit exact de quantités quelque peu perturbées.

3.7 Le calcul de la norme $\|x\|_2$

On donne maintenant un algorithme stable pour calculer la norme $\|x\|_2$ et l'on borne l'erreur.
Supposons l'existence d'une fonction $\mathrm{fl}\left(\sqrt{\cdot}\right)$ telle que

$$\mathrm{fl}(\sqrt{x}\,) = \sqrt{x}\,(1+\epsilon), \qquad |\epsilon| \leq \mathbf{u}. \tag{3.15}$$

On remarque tout d'abord que si

$$|x_i| < m = \beta^{L-1}$$

où m est le plus petit nombre positif dans $\mathcal{F}$ représentable sur l'ordinateur, alors $\mathrm{fl}\left(x_i^2\right) = 0$ et

$$\sqrt{\sum_{i=1}^{n} \mathrm{fl}\left(x_i^2\right)} = 0;$$

par conséquent, le simple algorithme

$$\sqrt{\sum x_i^2} \tag{3.16}$$

n'est pas acceptable si $|x_i| < m^{1/2}$, parce que l'erreur relative serait très grande. On remplace donc (3.16) par la formule équivalente:

Algorithme 3.2 *L'algorithme de la norme $\|x\|_2$:*

$$\alpha = \max_{1 \leq i \leq n} |x_i|, \tag{3.17}$$

$$\|x\|_2 = \begin{cases} 0 & \alpha = 0, \\ \alpha\sqrt{\sum_{i=1}^{n}(x_i/\alpha)^2} & \alpha \neq 0. \end{cases} \tag{3.18}$$

Estimons l'erreur de cette formule. Posons:

$$y_i = \mathrm{fl}\left(\left[\frac{x_i}{\alpha}\right]^2\right) = \left[\frac{x_i}{\alpha}(1+\epsilon_i')\right]^2 (1+\epsilon_i''), \qquad |\epsilon_i'|, |\epsilon_i''| < \mathbf{u},$$

$$z = \mathrm{fl}\left(\sum_{i=1}^{n} y_i\right) = \sum_{i=1}^{n} y_i(1+\eta_i), \qquad |\eta_i| < n\mathbf{u},$$
$$u = \mathrm{fl}\left(\sqrt{z}\right) = \sqrt{z}(1+\epsilon), \qquad |\epsilon| < \mathbf{u},$$
$$v = \mathrm{fl}(\alpha u) = \alpha u(1+\eta), \qquad |\eta| < \mathbf{u}.$$

On a donc:

$$\begin{aligned}
\mathrm{fl}\big(\|x\|_2\big) &:= (1+\eta)(1+\epsilon)\sqrt{\sum_{i=1}^{n} x_i^2(1+\epsilon_i')^2(1+\epsilon_i'')(1+\eta_i)} \\
&\leq (1+\mathbf{u})(1+\mathbf{u})\sqrt{\sum x_i^2(1+\mathbf{u})^2(1+\mathbf{u})(1+n\mathbf{u})} \\
&\leq (1+\mathbf{u})^2\sqrt{\sum x_i^2\left[1+(3+n)\mathbf{u}+(3+3n)\mathbf{u}^2+O\left(\mathbf{u}^3\right)\right]} \\
&\leq (1+\mathbf{u})^2\sqrt{\sum x_i^2\left[1+2\left(\frac{5+n}{2}\right)\mathbf{u}+\left(\frac{5+n}{2}\right)^2\mathbf{u}^2+O\left(\mathbf{u}^3\right)\right]} \\
&= (1+\mathbf{u})^2\sqrt{\sum x_i^2}\sqrt{\left(1+\frac{5+n}{2}\mathbf{u}\right)^2+O\left(\mathbf{u}^3\right)} \\
&= \sqrt{\sum x_i^2}\left[(1+\mathbf{u})^2\left(1+\frac{5+n}{2}\mathbf{u}\right)+O\left(\mathbf{u}^{3/2}\right)\right] \\
&= \|x\|_2\left(1+\frac{n+9}{2}\mathbf{u}\right)+O\left(\mathbf{u}^{3/2}\right).
\end{aligned}$$

De fait, une analyse plus serrée donne $(n+6)/4$ au lieu de $(n+9)/2$. Donc l'algorithme ci-dessus est stable et il admet une bonne erreur relative même si les $|x_i|$ sont petits. Si $n\mathbf{u}$ est suffisamment petit et si l'on accumule le produit scalaire, on obtient une borne aussi petite qu'en (3.10).

Exercice 3.2

(a) Démontrer le corollaire 3.2.

(b) Soit $E \in \mathbf{R}^{m\times n}$ $(m \geq n)$. Montrer:

$$\| \, |E| \, \|_2 \leq \sqrt{n}\, \|E\|_2.$$

(c) Montrer, par un exemple, que l'erreur du produit de deux matrices d'ordre 2 en arithmétique finie (t borné) peut être importante.

(d) Soit B une matrice régulière. Montrer, au moyen de (3.13), que

$$\frac{\|\,\mathrm{fl}(AB) - AB\|_F}{\|AB\|_F} \leq n\mathbf{u}\kappa_F(B) + O(\mathbf{u}^2).$$

Chapitre 4

Décomposition des matrices en facteurs

4.1 Réflexions de Householder

Définition 4.1 *Soit* $v \in \mathbf{R}^n$. *On nomme* réflexion de Householder *la matrice* $P \in \mathbf{R}^{n\times n}$*:*

$$P = I - 2\frac{vv^T}{v^T v}$$

et vecteur de Householder *le vecteur* v.

On voit que le $2^{\text{ième}}$ terme de P (sans le facteur -2) appliqué sur un vecteur x est la projection de x sur v:

$$\text{Proj}_v x = \frac{v^T x}{v^T v} v,$$

qui est indépendante de la longueur de v.
On voit facilement que P est symétrique et orthogonale. De plus,

$$Pv = v - 2v = -v$$

et

$$Pu = u \qquad \text{si } u^T v = 0.$$

Il suit de ces deux relations que P applique tout vecteur de $\text{lin}\{v\}$ dans son opposé et tout vecteur du supplémentaire orthogonal de $\text{lin}\{v\}$ en lui-même. Alors, Px est la réflexion de x dans l'hyperplan $\text{lin}\{v\}^\perp \in \mathbf{R}^{n-1}$ (v. fig. 4.1).

On peut voir, au moyen de l'identité:

$$\text{dét}\left(I_n + xy^T\right) = 1 + x^T y, \qquad x, y \in \mathbf{R}^n$$

que dét $P = -1$.

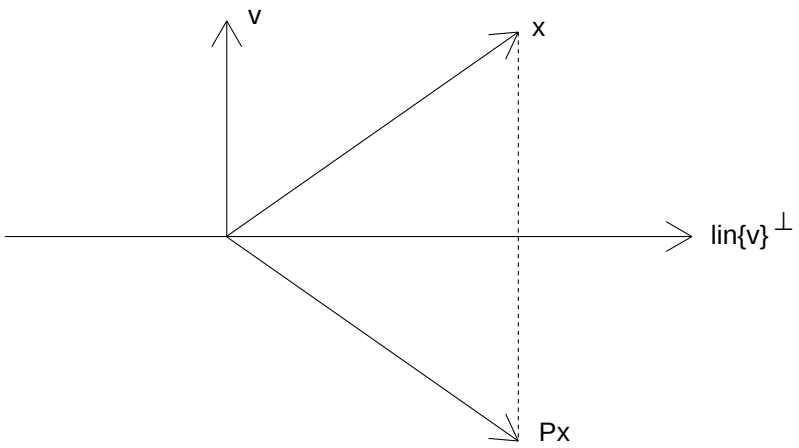

Figure 4.1: La réflexion Px de Householder.

Une réflexion de Householder est une modification de rang 1 de l'identité; elle sert à annuler un bloc de composantes d'un vecteur.

Par exemple, soit $x \in \mathbf{R}^n$ non nul. Pour annuler toutes les composantes de x sauf la première, on projette x sur e_1, c'est-à-dire on trouve v tel que $Px \in \text{lin}\{e_1\}$:

$$Px = \left(I - \frac{2vv^T}{v^T v}\right) x = x - \frac{2v^T x}{v^T v} v \in \text{lin}\{e_1\}.$$

On a donc nécessairement

$$v \in \text{lin}\{x, e_1\}, \quad \text{c'est-à-dire } v = x + \alpha e_1.$$

Alors

$$v^T x = x^T x + \alpha x_1$$

et

$$v^T v = x^T x + 2\alpha x_1 + \alpha^2;$$

donc

$$Px = \left(1 - 2\frac{x^T x + \alpha x_1}{x^T x + 2\alpha x_1 + \alpha^2}\right) x - 2\alpha \frac{v^T x}{v^T v} e_1.$$

Le coefficient de x s'annule si $\alpha = \pm ||x||_2$. Alors

$$v = x \pm ||x||_2 e_1 \quad \Rightarrow \quad Px = \left(I - 2\frac{vv^T}{v^T v}\right) x = \mp ||x||_2 e_1.$$

Les réflexions de Householder tirent leur utilité de cette simple détermination de v.

Exemple 4.1 *Soient* $x = (3,1,5,1)^T$ *et* $v = (9,1,5,1)^T$*; alors*

$$P = I - 2\frac{vv^T}{v^Tv} = \frac{1}{54}\begin{bmatrix} -27 & -9 & -45 & -9 \\ -9 & 53 & -5 & -1 \\ -45 & -5 & 29 & -5 \\ -9 & -1 & -5 & 53 \end{bmatrix}$$

est telle que $Px = (-6,0,0,0)^T$.

Pour réduire l'erreur relative dans le facteur

$$\beta = 2/v^Tv,$$

on choisit le signe de α,

$$\alpha = \operatorname{sgn}(x_1)||x||_2,$$

qui maximise $||v||_2$ et donne $||v||_2 \geq ||x||_2$; alors

$$v = x + \operatorname{sgn}(x_1)||x||_2 e_1.$$

De plus, pour éviter les débordements dans le calcul de $||x||_2$ on part de $x/||x||_\infty$, c'est-à-dire on emploie l'algorithme stable du chapitre précédent.

En général, pour annuler les $(j-k)$ composantes successives de x, $x_{k+1}, \ldots, x_j$, on prend

$$\alpha^2 = x_k^2 + x_{k+1}^2 + \cdots + x_j^2;$$

alors

$$v = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_k + \operatorname{sgn}(x_k)\alpha \\ x_{k+1} \\ \vdots \\ x_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad Px = \begin{bmatrix} x_1 \\ \vdots \\ x_{k-1} \\ -\operatorname{sgn}(x_k)\alpha \\ 0 \\ \vdots \\ 0 \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix},$$

où

$$P = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & \tilde{P} & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix} = \operatorname{diag}(I_{k-1}, \tilde{P}, I_{n-j})$$

et $\tilde{P}$ est donnée comme au début avec $\tilde{x} = (x_k, x_{k+1}, \ldots, x_j)^T$.

L'algorithme suivant calcule v^T et β.

Algorithme 4.1 *Soient $x \in \mathbf{R}^n$ non nul et $1 \leq k < j \leq n$. L'algorithme calcule*

$$v = (0, \ldots, 0, v_k, \ldots, v_j, 0, \ldots, 0)^T$$

et

$$\beta = \frac{2}{v^T v}$$

tels que les composantes de $k+1$ à j de

$$\left(I - \beta v v^T\right) x$$

soient nulles.

$m := \max\{|x_k|, \ldots, |x_j|\}$
$\alpha := 0$
Pour $i = k$ *à* j
  $v_i := x_i/m$
  $\alpha := \alpha + v_i^2$
$\alpha := \sqrt{\alpha}$
$\beta := 1/[\alpha(\alpha + |v_k|)]$
$v_k := v_k + \mathrm{sgn}(v_k)\alpha$ □

L'algorithme requiert environ $2(j-k)$ flops.

On vérifie qu'on obtient bien la valeur de β:

$$\frac{2}{v^T v} = \frac{2}{[v_k + \mathrm{sgn}(v_k)\alpha]^2 + v_{k+1}^2 + \cdots + v_j^2} = \frac{2}{2|v_k|\alpha + 2\alpha^2} = \frac{1}{\alpha\left(\alpha + |v_k|\right)}.$$

Il est important d'exploiter la structure de la transformation de Householder dans les calculs, c'est-à-dire:

$$\begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & \tilde{P} & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix} \begin{bmatrix} A_1 & A_2 & A_3 \\ A_4 & A_5 & A_6 \\ A_7 & A_8 & A_9 \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & A_3 \\ \tilde{P}A_4 & \tilde{P}A_5 & \tilde{P}A_6 \\ A_7 & A_8 & A_9 \end{bmatrix}.$$

On voit que seule la deuxième ligne bloc est modifiée par $\tilde{P}$; c'est ce que fait l'algorithme suivant.

Algorithme 4.2 *Soit $A \in \mathbf{R}^{n\times q}$, $v = (0, \ldots, 0, v_k, \ldots, v_j, 0, \ldots, 0)^T$ et $\beta = 2/v^T v$, alors l'algorithme renouvelle A par $\left(I - \beta v v^T\right) A$.*

Pour $p = 1, \ldots, q$
  $s := v_k a_{kp} + \cdots + v_j a_{jp}$
  $s := \beta s$
  Pour $i = k, \ldots, j$
    $a_{ip} := a_{ip} - s v_i$ □

L'algorithme requiert environ $2q(j-k+1)$ flops.

L'analyse de l'erreur rétrograde des deux algorithmes qui précèdent montre leur fiabilité. En effet, soient $\hat{v}$ et $\hat{\beta}$ les valeurs calculées de v et β, et posons

$$\widehat{P} = I - \hat{\beta}\hat{v}\hat{v}^T.$$

Alors $\widehat{P}$ diffère peu de la valeur exacte $P = I - \beta vv^T$, comme l'a montré Wilkinson:

$$||P - \widehat{P}||_2 \leq 10\mathbf{u}.$$

De plus, le renouvellement $\mathrm{fl}(\widehat{P}A)$ diffère peu de PA:

$$\mathrm{fl}(\widehat{P}A) = PA + E,$$

où

$$||E||_2 \leq c(j-k+1)^2\mathbf{u}||A||_2, \qquad c = O(1).$$

En pratique, si Q est le produit de matrices de Householder:

$$Q = P_1P_2\cdots P_r, \qquad P_i = I - \beta_i v^{(i)}\left[v^{(i)}\right]^T,$$

pour des raisons d'économie de flops et de mémoire, on garde souvent en mémoire non pas Q mais ses facteurs sous la forme β_i et $v^{(i)}$. Par exemple, dans plusieurs algorithmes on renouvelle $A \in \mathbf{R}^{n\times q}$ par QA en $2nrq$ flops dans l'ordre

$$[P_1\{P_2\cdots[P_{r-1}(P_rA)]\cdots\}],$$

selon l'algorithme suivant:

Pour $i = r, \ldots, 1$
    $A := P_iA$

alors que le renouvellement dans l'ordre opposé: $[\{\cdots(P_1P_2)\cdots P_{r-1}\}P_r]A$ selon l'algorithme suivant:

$Q := I$
Pour $i = 1, \ldots, r$
    $Q := QP_i$
$A := QA$

requiert $2n^2r + n^2q$ flops.
S'il faut accumuler les P_i, on le fait aussi dans l'ordre indiqué plus haut:

$Q := I$
Pour $i = r, \ldots, 1$
    $Q := P_iQ$

pour tirer avantage de la structure des produits partiels

$$P_{i+1}\cdots P_r = \mathrm{diag}\left(I_i, \widehat{Q}\right),$$

où $\widehat{Q} \in \mathbf{R}^{(n-i)\times(n-i)}$. L'accumulation dans l'ordre opposé produit un matrice pleine dès la première passe de la boucle $Q := QP_i$.

Exercice 4.1

(a) Exécuter les algorithmes 4.1 et 4.2 avec $k = 2$,

$$x = \begin{bmatrix} 1 \\ 7 \\ 2 \\ 3 \\ -1 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & -1 \\ 7 & -1 \\ 2 & 0 \\ 3 & 1 \\ -1 & 2 \end{bmatrix}.$$

(b) Soient x et y deux vecteurs non nuls de $\mathbf{R}^n$. Trouver une matrice de Householder P telle que Px soit un multiple de y.

(c) Soient $A \in \mathbf{R}^{n\times q}$ et $P \in \mathbf{R}^{q\times q}$ une matrice de Householder. Donner un algorithme qui fait le renouvellement $A := AP$.

(d) Soit $x \in \mathbf{C}^n$ non nul et écrivons $x_1 = |x_1|e^{i\theta}$, $\theta \in \mathbf{R}$. Posons

$$u = x + e^{i\theta}\|x\|_2 e_1$$

et

$$P = I - \frac{2}{u^H u} uu^H.$$

Montrer que P est unitaire, c'est-à-dire $P^{-1} = P^H$ où $P^H := \overline{P}^T$, et que

$$Px = -e^{i\theta}\|x\|_2 e_1.$$

(e) Soient x et y deux vecteurs non nuls de $\mathbf{R}^n$. Empoyer deux réflexions de Householder pour montrer que $\operatorname{rg} xy^T = 1$.

(f) Soient $x, y \in \mathbf{R}^n$. Montrer, au moyen de réflexions de Householder:

$$\det(I_n + xy^T) = 1 + x^T y.$$

4.2 Rotations de Givens

On emploie une rotation de Givens pour annuler un seul élément à la fois, par opposition à une réflexion de Householder qui annule une colonne d'éléments. Les rotations en questions sont des modifications de rang 2 de l'identité de la forme:

$$G(i,k,\theta) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{matrix} \\ \\ i \\ \\ k \\ \\ \end{matrix}$$

$$\begin{matrix} & i & & k \end{matrix}$$

où $c = \cos\theta$ et $s = \sin\theta$ et l'angle de la rotation dans le plan (i,k) est θ. Il est clair que les rotations de Givens sont données par des matrices orthogonales.

L'application de $G(i,k,\theta)^T$ sur $x \in \mathbf{R}^n$ donne

$$\begin{aligned} y_i &= cx_i - sx_k \\ y_k &= sx_i + cx_k \\ y_j &= x_j \qquad\qquad j \neq i,k. \end{aligned}$$

On annule y_k en prenant

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \qquad s = \frac{-x_k}{\sqrt{x_i^2 + x_k^2}}.$$

L'algorithme suivant évite les dépassements.

Algorithme 4.3 *Soient les scalaires a et b; l'algorithme calcule $c = \cos\theta$ et $s = \sin\theta$ tels que*

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}.$$

Si $b = 0$
  $c = 1; s = 0$
sinon
  si $|b| > |a|$
    $\tau = -a/b;\ s = 1/\sqrt{1+\tau^2};\ c = s\tau$
  sinon
    $\tau = -b/a;\ c = 1/\sqrt{1+\tau^2};\ s = c\tau$
  fin
fin □

L'algorithme requiert 4 flops et une racine carrée; il ne calcule pas θ et ne fait pas appel aux fonctions trigonométriques inverses.

Exemple 4.2 *Soient $x = (1,2,3,4,)^T$, $\cos\theta = 1/\sqrt{5}$ et $\sin\theta = -2/\sqrt{5}$; alors*

$$G(2,4,\theta)^T x = (1, \sqrt{20}, 3, 0)^T.$$

Il est important d'exploiter la structure simple de $G(i,k,\theta)$ dans la formation du produit $G(i,k,\theta)A$. On remarque que la prémultiplication affecte uniquement les lignes i et k et ce d'une façon très simple. L'algorithme qui suit, tient compte de ces remarques.

Algorithme 4.4 *Soient $A \in \mathbf{R}^{n\times q}$, $c = \cos\theta$, $s = \sin\theta$, $1 \leq i < k \leq n$. Alors l'algorithme renouvelle A par $G(i,k,\theta)^T A$.*

Pour $j = 1, \ldots, q$
  $v := a_{ij}$
  $w := a_{kj}$
  $a_{ij} := cv - sw$
  $a_{kj} := sv + cw$ □

L'algorithme requiert $4q$ flops.

Les propriétés numériques des rotations de Givens sont excellentes. Les valeurs calculées de c et s par l'algorithme précédent satisfont les estimations suivantes:

$$\hat{c} = c(1+\epsilon_c), \qquad \epsilon_c = O(\mathbf{u}),$$

$$\hat{s} = s(1+\epsilon_s), \qquad \epsilon_s = O(\mathbf{u}).$$

L'emploi de $\hat{c}$ et $\hat{s}$ dans l'algorithme précédent donne

$$\text{fl}\left[\widehat{G}(i,k,\theta)^T A\right] = G(i,k,\theta)^T A + E,$$

où

$$\frac{||E||_2}{||A||_2} = O(\mathbf{u}).$$

Exercice 4.2

(a) Soit $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $x_1, x_2 \in \mathbf{C}$. Trouver une matrice unitaire

$$Q = \begin{bmatrix} c & \bar{s} \\ -s & c \end{bmatrix}, \qquad c \in \mathbf{R},\ c^2 + |s|^2 = 1,$$

telle que la $2^{\text{ième}}$ composante de $Q^H x$ soit nulle.

(b) Soient deux vecteurs unités $x, y \in \mathbf{R}^n$. Utiliser les rotations de Givens pour trouver une matrice orthogonale Q telle que $Q^T x = y$.

(c) Trouver $c = \cos\theta$ et $s = \sin\theta$ tels que

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} 5 \\ 12 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \alpha \text{ un scalaire.}$$

4.3 Transformations de Gauss

On peut annuler des éléments d'un vecteur par des transformations qui ne sont pas orthogonales. Par exemple, si $x_1 \neq 0$ et $\tau = -x_2/x_1$, alors

$$\begin{bmatrix} 1 & 0 \\ \tau & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}.$$

D'une façon générale, soit $x \in \mathbf{R}^n$ avec $x_k \neq 0$ et soit le *vecteur de Gauss*

$$\tau = (0, \ldots, 0, \tau_{k+1}, \ldots, \tau_n)^T, \qquad \tau_i = \frac{x_i}{x_k}, \quad i = k+1, \ldots, n.$$

Alors la *transformation de Gauss*

$$M_k = I - \tau e_k^T$$

annule les $x_i, i = k+1, \ldots, n$:

$$M_k x = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & & 1 & 0 & & 0 \\ 0 & & -\tau_{k+1} & 1 & & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\tau_n & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

La matrice M_k est triangulaire inférieure. Elle est un perturbation de rang 1 de l'identité. Les composantes $\tau_{k+1}, \ldots, \tau_n$ sont appelées *multiplicateurs*.

On donne deux algorithmes, soit pour le calcul des multiplicateurs et le produit $M_k A$.

Algorithme 4.5 *Soit $x \in \mathbf{R}^n$ avec $x_k \neq 0$. L'algorithme calcule*

$$\tau = (0, \ldots, 0, \tau_{k+1}, \ldots, \tau_n)^T$$

tel que les composantes $k+1, \ldots, n$ de $(I - \tau e_k^T)x$ soient nulles.

Pour $i = k+1, \ldots, n$
  $\tau_i := x_i / x_k$ □

L'argorithme requiert $n - k$ flops.

La multiplication par une matrice de Gauss $M_k = I - \tau e_k^T$ est toute simple:

$$M_k x = x - x_k \tau.$$

Le renouvellement de A par $M_k A$ est aussi très simple.

Algorithme 4.6 *Soient $A \in \mathbf{R}^{n \times q}$ et $\tau = (0, \ldots, 0, \tau_{k+1}, \ldots, \tau_n)^T$. L'algorithme renouvelle A par $(I - \tau e_k^T)A$.*

Pour $j = 1, \ldots, q$
  Pour $i = k+1, \ldots, n$
    $a_{ij} := a_{ij} - a_{kj}\tau_i$ □

L'argorithme requiert $(n - k)q$ flops.

Exemple 4.3 *Soit*

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad \textit{et } \tau = \begin{bmatrix} 0 \\ 0 \\ 3/2 \\ 2 \end{bmatrix}, \quad \textit{alors } \left(I - \tau e_2^T\right) x = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix},$$

Remarque 4.1 L'algorithme précédent requiert la moitié du travail exigé par l'algorithme 4.2 de Householder pour renouveler A. Le prix de cette économie est une perte de stabilité si le vecteur de Gauss devient grand. En effet, l'algorithme 4.5 donne la valeur numérique:

$$\hat{\tau} = \tau + e, \qquad |e| \leq \mathbf{u}|\tau|,$$

et l'algorithme 4.6 donne le renouvellement numérique

$$\text{fl}\left[\left(I - \hat{\tau}e_k^T\right)A\right] = \left(I - \tau e_k^T\right)A + E, \quad |E| \leq 3\mathbf{u}\left[|A| + |\tau|\ \left|a_k^T\right|\right] + O\left(\mathbf{u}^2\right),$$

avec $a_k = (a_{k1}, \ldots, a_{kq})^T$. On voit que l'erreur peut être grande si τ est grand devant $|A|$, d'où la nécessité du pivotage avec la décomposition LU.

On termine avec quelques remarques utiles. Soit

$$M = M_k \cdots M_1,$$

où

$$M_i = I_n - \tau^{(i)} e_i^T$$

avec

$$\tau^{(i)} = (0, \ldots, -l_{i+1,i}, \ldots, -l_{ni})^T.$$

On voit que M est triangulaire inférieure. En pratique il est utile de conserver M en facteurs pour économiser la mémoire si $k < n$; par exemple,

$$\left[-\tau^{(1)}, -\tau^{(2)}\right] = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \\ l_{31} & l_{32} \\ \vdots & \vdots \\ l_{n1} & l_{n2} \end{bmatrix}, \qquad k = 2.$$

Cette représentation de M équivaut à emmagasiner les k premières colonnes de $M^{-1} = M_1^{-1} \cdots M_k^{-1}$ du fait que

$$M_i^{-1} = I + \tau^{(i)} e_i^T$$

et $e_j^T \tau^{(i)} = 0$ pour $j \leq i$. Alors

$$M^{-1} = \left(I + \tau^{(1)} e_1^T\right) \cdots \left(I + \tau^{(k)} e_k^T\right) = I + \sum_{i=1}^{k} \tau^{(i)} e_i^T.$$

Presque tous les algorithmes qui emploient la transformation de Gauss, exploitent cette propriété.

Exercice 4.3

(a) Soit une transformation de Gauss $M_k = I - \alpha e_k^T$. Montrer que

$$M_k^{-1} = I + \alpha e_k^T.$$

(b) Trouver une transformation de Gauss M telle que

$$M \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \\ 8 \end{bmatrix}.$$

Chapitre 5

Élimination gaussienne

5.1 Décomposition LU

5.2 Analyse de l'erreur

5.3 Pivotage

Soit $A \in \mathbf{R}^{n\times n}$ régulière. L'élimination gaussienne simple fait la décomposition LU de A, L triangulaire inférieure avec $l_{ii} = 1$ et U triangulaire supérieure. On a

$$M_{n-1} \cdots M_2 M_1 A = U,$$

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U =: LU.$$

Alors

$$Ux = M_{n-1} \cdots M_2 M_1 b$$

et x est obtenu par substitution avant, $Ly = b$, (v. G–VL, l'algorithme 4.1-1) et substitution arrière, $Ux = y$ (v. G–VL, l'algorithme 4.1-2).
L'analyse de l'erreur rétrograde donne

$$\widehat{L}\widehat{U} = A + E, \qquad (A + E)\hat{x} = b,$$

où

$$|E| \leq n\mathbf{u}[3|A| + 5|\widehat{L}|\,|\widehat{U}|] + O\left(\mathbf{u}^2\right).$$

Si un pivot est petit, $|\widehat{L}|$ peut être grande en norme et l'erreur peut être grande. L'algorithme 4.2-1 (v. G–VL) fait la décomposition LU sans pivotage en $n^3/3$ flops.

L'élimination gaussienne avec pivotage total permute les colonnes et les lignes de A pour mettre le pivot maximum en module dans le coin gauche supérieur

$a_{kk}^{(k)}$ de la matrice partielle à la $k^{\text{ième}}$ étape. A la $(n-1)^{\text{ième}}$ étape, on obtient la matrice triangulaire supérieure

$$M_{n-1}P_{n-1}\cdots M_2P_2M_1P_1A\Pi_1\Pi_2\cdots\Pi_{n-1} = U.$$

Ecrivons

$$P = P_{n-1}\cdots P_1, \qquad \Pi = \Pi_1\cdots\Pi_{n-1}$$

et posons

$$L = P(M_{n-1}P_{n-1}\cdots M_2P_2M_1P_1)^{-1}.$$

On peut voir que L est triangulaire inférieure avec $l_{ii} = 1$ et $PA\Pi = LU$.

Le système $Ax = b$ devient donc

$$(PA\Pi)(\Pi^{-1}x) = LU(\Pi^{-1}x) = Pb$$

qu'on résout au moyen des étapes suivantes:

$Ly = Pb$ produit y par subsitution avant,
$Uz = y$ produit z par subsitution arrière,
$x = \Pi z$.

L'analyse de l'erreur rétrograde donne

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

où

$$|E| \le n\mathbf{u}\{3|A| + 5\widehat{P}^T|\widehat{L}|\,|\widehat{U}|\widehat{\Pi}^T\} + O\left(\mathbf{u}^2\right).$$

Puisque $|l_{ij}| \le 1$, alors $\|\widehat{L}\|_\infty \le n$. Il est évident que $\|\widehat{P}^T\|_\infty = 1$ et $\|\widehat{\Pi}^T\|_\infty = 1$. Alors

$$\|E\|_\infty \le n\mathbf{u}\{3\|A\|_\infty + 5n\|\widehat{U}\|_\infty\} + O\left(\mathbf{u}^2\right).$$

L'algorithme 4.4-1 (v. G–VL) fait la décomposition LU avec pivotage complet en $n^3/3$ flops et $n^3/3$ comparaisons.

Un analyse de la stabilité de l'élimination gaussienne fait intervenir le *facteur de croissance*

$$\rho := \max_{i,j,k} \frac{|\hat{a}_{ij}^{(k)}|}{\|A\|_\infty},$$

où

$$A^{(k)} = M_kP_k\cdots M_1P_1A\Pi_1\cdots\Pi_k.$$

Alors

$$\|E\|_\infty \le 8n^3\rho\|A\|_\infty\mathbf{u} + O\left(\mathbf{u}^2\right).$$

Le facteur n^3 est très pessimiste; en pratique on peut prendre $\rho \approx 10$. Donc l'éliminatation gaussienne avec pivotage complet est stable, c'est-à-dire

$$(A+E)\hat{x} = b + e, \qquad \|E\|_\infty \le \mathbf{u}\|A\|_\infty, \|e\|_\infty \le \mathbf{u}\|b\|_\infty.$$

L'élimination gaussienne avec pivotage partiel permute les lignes seulement; donc $\Pi_i = I$ et $\Pi = I$. Elle donne donc la matrice triangulaire supérieure

$$M_{n-1}P_{m-1}\cdots M_1P_1A = U.$$

Si l'on écrit

$$P = P_{n-1}\cdots P_1, \qquad L = P(M_{n-1}P_{n-1}\cdots M_1PL_1)^{-1},$$

on a

$$PA = LU.$$

On résout $Ax = b$ sous la forme $PAx = Pb$:

$$Ly = Pb, \qquad x = Uy.$$

L'algorithme 4.4-2 (v. G.VL) fait la décomposition LU avec pivotage partiel en $n^3/3$ flops et $O\left(n^2\right)$ comparaisons.

Il existe des exemples farfelus avec pivotage partiel où le facteur de croissance $\rho \approx 2^n$. Mais en pratique, ρ a le même ordre de croissance avec pivotage partiel qu'avec pivotage complet. Le pivotage partiel est donc une méthode fiable.

5.4 Affinage de la solution

Si l'on résout $Ax = b$ par Gauss avec pivotage, la solution numérique $\hat{x}$ satisfait le système perturbé

$$(A + E)\hat{x} = b, \qquad \frac{\|E\|_\infty}{\|A\|_\infty} \cong \beta^{-t},$$

où

$$\|E\|_\infty \leq 8n^3\rho\|A\|_\infty\mathbf{u} + O\left(\mathbf{u}^2\right).$$

Par le théorème 2.2, si $\|E\| \leq \delta\|A\|$ et $\|E\|\,\|A^{-1}\| = r < 1$, alors

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \frac{2\delta\kappa(A)}{1 - r};$$

donc

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \cong \beta^{-t}\kappa_\infty(A)$$

parce que $2/(1 - r) \cong 1$. Alors

$$\|b - A\hat{x}\|_\infty = \|E\hat{x}\|_\infty \cong \beta^{-t}\|A\|_\infty\|x\|_\infty$$

et si $\kappa_\infty(A) \cong \beta^q$,

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \cong \beta^{q-t}.$$

Conclusions probables: si $\hat{x}$ est la solution obtenue par l'élimination gaussienne, on peut dire que

Table 5.1: Effet des erreurs de chute pour l'élimination gaussienne avec pivotage partiel.

β	t	$\hat{x}_1$	$\hat{x}_2$	$\frac{\|\hat{x}-x\|_\infty}{\|x\|_\infty}$	$\frac{\|b-A\hat{x}\|_\infty}{\|A\|_\infty\|\hat{x}\|_\infty}$
10	3	2.11	-3.17	5×10^{-2}	2.0×10^{-3}
10	4	1.986	-2.975	8×10^{-3}	1.5×10^{-4}
10	5	2.001 9	$-3.003\,2$	1×10^{-3}	2.1×10^{-6}
10	6	2.000 25	$-3.000\,94$	3×10^{-4}	4.2×10^{-7}

(a) le résidu $b - A\hat{x}$ est petit, et

(b) $\hat{x}$ a s décimales correctes, où

$$10^{-s} = \beta^{-t}\kappa_\infty(A),$$

c'est-à-dire

$$s = \log_{10}\beta^t - \log_{10}(\kappa_\infty(A)).$$

Exemple. Soit le système $Ax = b$ et la solution exacte x:

$$\begin{bmatrix} 0.986 & 0.059 \\ 0.409 & 0.237 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.235 \\ 0.107 \end{bmatrix}, \qquad x = \begin{bmatrix} 2 \\ -3 \end{bmatrix}.$$

Le conditionnement $\kappa_\infty(A) \approx 700$. La table suivante résume l'effet des erreurs de chute de l'élimination gaussienne avec pivotage partiel. Les résultats obtenus confirment les conclusions probables (a) et (b).

On affine la solution par l'itération suivante où l'on calcule le résidu $r = b - Ax$ en précision double (avec $2t$ chiffres).

Soit x la solution de $LUx = b$ avec t chiffres. Alors, on répète l'algorithme suivant:

$$\begin{array}{rrcll} & r & := & b - Ax, & 2t \text{ chiffres} \\ \text{Résoudre} & Ly & = & r, & t \text{ chiffres} \\ \text{Résoudre} & Uz & = & y, & t \text{ chiffres} \\ & x & := & x + z, & t \text{ chiffres} \end{array}$$

On remarque l'emploi de A et non de décomposition LU dans le calcul du résidu en précision double. En effet, L et U contiennent des erreurs. Le coût de l'algorithme est de l'ordre de $O\left(n^2\right)$ ce qui se compare bien avec le coût de la décomposition LU initiale qui est de l'ordre de $O\left(n^3\right)$.

Exemple 5.1 *Soit $\beta = 10$ et $t = 3$. On affine la solution du système de l'exemple précédent:*

$$\begin{bmatrix} 0.986 & 0.059 \\ 0.409 & 0.237 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.235 \\ 0.107 \end{bmatrix}, \qquad x = \begin{bmatrix} 2 \\ -3 \end{bmatrix}.$$

où on a indiqué la solution exacte. L'algorithme d'affinage donne:

$$\hat{x} = \begin{bmatrix} 2.11 \\ -3.17 \end{bmatrix}, \begin{bmatrix} 1.99 \\ -2.99 \end{bmatrix}, \begin{bmatrix} 2.00 \\ -3.00 \end{bmatrix}.$$

On fait le raisonnement heuristique suivant. Soit le conditionnement de A, $\kappa_\infty(A) \approx \beta^p$ en base β. Alors, après k affinages, le nombre approximatif de chiffres corrects dans la solution x sera

$$\min\{t, k(t-q)\}.$$

Si $t - q \leq 0$, on ne s'attend pas à une amélioration de la solution.

On remarque que l'algorithme d'affinage dépend de la précision de la machine. De plus, il faut garder une copie de A en mémoire.

Exercice 5.1

(a) Soit $\beta = 10$, $t = 2$ et l'on tronque. Résoudre

$$\begin{bmatrix} 11 & 15 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \end{bmatrix}$$

au moyen de l'élimination gaussienne avec pivotage partiel et affiner la solution une fois.

Chapitre 6

Systèmes particuliers

6.1 Décompositions LDM^T et LDL^T

On réarrange la décomposition LU en décomposition LDM^T où L et M sont triangulaires inférieures et D est diagonale.

Théorème 6.1 *Soit $A \in \mathbf{R}^{n\times n}$. Si tous les mineurs principaux de A sont non nuls, alors*

$$A = LDM^T,$$

où

$$l_{ii} = 1, \qquad m_{ii} = 1,$$

L et M sont triangulaires inférieures et D est diagonale. De plus, L, M et D sont uniques.

Démonstration. Etant donné la décomposition LU de A, on a

$$A = LU =: LD(D^{-1}U) =: LDM^T,$$

où $D = \mathrm{diag}(u_{ii})$ et $M = (D^{-1}U)^T$. □

Remarque 6.1 Soit $Ax = b$. Si L, D, M^T sont connues, alors

$$\begin{aligned} Ly &= b, && n^2/2 \quad \text{flops},\\ Dz &= y, && n \quad \text{flops},\\ M^T x &= z, && n^2/2 \quad \text{flops}. \end{aligned}$$

Algorithme 6.1 *Soit $A \in \mathbf{R}^{n\times n}$; l'algorithme calcule L, D, M^T; $l_{ij}, i > j$, et $m_{ij}, i < j$, renouvellent a_{ij}. L'algorithme s'arrête si la décomposition n'existe pas.*

Pour $k = 1, \ldots, n$
  Pour $p = 1, \ldots, k-1$

$r_p := d_p a_{pk}$
$w_p := a_{kp} d_p$
$d_k := a_{kk} - \sum_{p=1}^{k-1} a_{kp} r_p$
Si $d_k = 0$
alors
sortir
sinon
Pour $i = k+1, \ldots, n$
$a_{ik} := \left(a_{ik} - \sum_{p=1}^{k-1} a_{ip} r_p\right) \Big/ d_k$
$a_{ki} := \left(a_{ki} - \sum_{p=1}^{k-1} w_p a_{pi}\right) \Big/ d_k$ □

L'algorithme requiert $n^3/3$ flops. □

L'analyse de l'erreur rétrograde donne:

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

$$|E| \leq n\mathbf{u}[3|A| + 5|\hat{L}|\,|\hat{D}|\,|\hat{M}^T|] + O\left(\mathbf{u}^2\right).$$

En pratique, on pivote pour borner $\hat{L}$: $PA = LDM^T$.

Exemple 6.1

$$A = \begin{bmatrix} 10 & 20 & 30 \\ 20 & 45 & 80 \\ 30 & 80 & 171 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix}.$$

Notre but, en fait, est de considérer les matrices symétriques, $A^T = A$.

Théorème 6.2 *Soit A régulière et symétrique et sa décomposition $A = LDM^T$, alors $M = L$ et*

$$A = LDL^T.$$

Démonstration. Considérons les identités matricielles:

$$M^{-1}AM^{-T} = M^{-1}LDM^TM^{-T} = M^{-1}LD.$$

Puisque le premier membre est symétrique et le troisième est triangulaire inférieur, il s'ensuit que les trois membres sont diagonaux.
Puisque D est régulière, alors $M^{-1}L$ est diagonale, et puisque $M^{-1}L$ est triangulaire inférieure avec les éléments de la diagonale = 1, alors $M^{-1}L = I$, c'est-à-dire $M = L$. □

Algorithme 6.2 *Soit $A \in \mathbf{R}^{n\times n}$ symétrique; l'algorithme calcule L et D; $l_{ij}, i > j$, renouvellent a_{ij}. L'algorithme s'arrête si la décomposition n'existe pas.*

Pour $k = 1, \ldots, n$
Pour $p = 1, \ldots, k-1$

$r_p := d_p a_{pk}$
$d_k := a_{kk} - \sum_{p=1}^{k-1} a_{kp} r_p$
Si $d_k = 0$
alors
sortir
sinon
Pour $i = k+1, \ldots, n$
$a_{ik} := \left(a_{ik} - \sum_{p=1}^{k-1} a_{ip} r_p\right) \Big/ d_k$ □

L'algorithme requiert $n^3/6$ flops.

6.2 Systèmes définis positifs

Définition 6.1 *Une matrice* $A \in \mathbf{R}^{n\times n}$ *est* définie positive *si* $x^T Ax > 0, \forall x \neq 0$.

Théorème 6.3 *Soit A définie positive, alors A admet la décomposition* LDM^T *où les* $d_i > 0$.

Démonstration. Toutes les sous-matrices principales sont définies positives et donc régulières. Par le théorème à la section précédente, on peut trouver deux matrices triangulaires inférieures, L et M avec 1 sur la diagonale, et une matrice diagonale D, telles que

$$A = LDM^T.$$

Puisque

$$\begin{aligned} S &= DM^T L^{-T} && \text{triangulaire supérieure} \\ &= L^{-1} A L^{-T} && \text{définie positive,} \end{aligned}$$

et

$$s_{ii} = d_i,$$

alors

$$d_i > 0. \quad \square$$

Remarque 6.2 Même si la décomposition LDM^T existe, il n'est pas toujours indiqué de la calculer; en effet, les facteurs pourraient être très grands. Par exemple, pour

$$A = \begin{bmatrix} \epsilon & m \\ -m & \epsilon \end{bmatrix}, \qquad m \gg \epsilon > 0,$$

$$A = LDM^T = \begin{bmatrix} 1 & 0 \\ -m/\epsilon & 1 \end{bmatrix} \begin{bmatrix} \epsilon & 0 \\ 0 & \epsilon + m^2/\epsilon \end{bmatrix} \begin{bmatrix} 1 & m/\epsilon \\ 0 & 1 \end{bmatrix}.$$

Puisque $m/\epsilon \gg 1$, il serait bon de pivoter ici, quitte à perdre le caractère défini positif de A.
On vérifie facilement que A est définie positive:

$$x^T Ax = \epsilon \|x\|_2^2 > 0 \qquad \forall x \neq 0.$$

Le thórème qui suit donne une indication sur la croissance possible des éléments de la décomposition LDM^T.

Théorème 6.4 *Soit $A \in \mathbf{R}^{n\times n}$ définie positive. Notons respectivement*

$$T := (A + A^T)/2, \qquad S := (A - A^T)/2,$$

les parties symétrique et antisymétrique de A. Si

$$A = LDM^T$$

alors

$$\| \, |L| \, |D| \, |M^T| \, \|_F \leq n \left[\|T\|_2 + \left\| ST^{-1}S \right\|_2 \right].$$

Démonstration. Voir Golub-Van Loan (1979).

Remarque 6.3 Puisque $\|T\|_2 \leq \|A\|_2$, on voit qu'on peut sans crainte éviter de pivoter si le facteur

$$\Omega = \frac{\|ST^{-1}S\|_2}{\|A\|_2}$$

n'est pas trop grand; c'est-à-dire quand la norme de S est petite devant le conditionnement de T. Ceci est toujours le cas quand A est symétrique et définie positive; on a alors la décomposition de Cholesky: $A = GG^T$.

Théorème 6.5 *Soit $A = A^T$ définie positive. Alors il existe une matrice triangulaire inférieure G avec $g_{ii} > 0$ et telle que*

$$A = GG^T.$$

Démonstration. Par le théorème précédent, on a

$$A = LDL^T, \qquad d_i > 0.$$

Alors

$$G = LD^{1/2}$$

est triangulaire inférieure et

$$A = GG^T. \quad \square$$

Remarque 6.4 On voit que

$$A = \begin{bmatrix} 2 & -2 \\ -2 & 5 \end{bmatrix}$$

est définie positive puisque les valeurs propres $\lambda_1 = 1, \lambda_2 = 6$ sont positives. On a

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

Pour obtenir G on compare les composantes des deux membres de $A = GG^T$:

$$a_{ik} = \sum_{p=1}^{k} g_{ip} g_{kp}, \qquad i \geq k.$$

Si on réarrange cette expression, on obtient:

$$\begin{aligned} g_{ik} &= \left(a_{ik} - \sum_{p=1}^{k-1} g_{ip} g_{kp} \right) \Big/ g_{kk}, \qquad i > k, \\ g_{kk} &= \left(a_{kk} - \sum_{p=1}^{k-1} g_{kp}^2 \right)^{1/2}. \end{aligned}$$

Algorithme 6.3 (Cholesky-colonne) *Soit $A^T = A \in \mathbf{R}^{n \times n}$ définie positive. On calcule G triangulaire inférieure telle que $A = GG^T$ et g_{ij} $(i \geq j)$ remplace a_{ij}.*

Pour $k = 1, 2, \ldots, n$,
  $a_{kk} := \left(a_{kk} - \sum_{p=1}^{k-1} a_{kp}^2 \right)^{1/2}$
  Pour $i = k+1, \ldots, n$,
    $a_{ik} := \left(a_{ik} - \sum_{p=1}^{k-1} a_{ip} a_{kp} \right) \Big/ a_{kk}$ □

L'algorithme requiert $n^3/6$ flops.

Wilkinson a démontré que la solution numérique $\hat{x}$ de $Ax = b$ obtenue au moyen de la décomposition $A = GG^T$ et de la résolution de $Gy = b$ et de $G^T x = y$, satisfait:

$$(A + E)\hat{x} = b, \qquad \|E\|_2 \leq c_n \mathbf{u} \|A\|_2,$$

où c_n est une petite constante dépendant de la dimension n. De plus, si

$$q_n \mathbf{u} \|A\|_2 \|A^{-1}\|_2 \leq 1,$$

où q_n est une autre petite constante, on peut compléter la décomposition de Cholesky sans qu'on prenne la racine carrée de nombres négatifs.

Exemple 6.2 *Appliquons l'algorithme précédent à la matrice définie positive:*

$$A = \begin{bmatrix} 100 & 15 & .01 \\ 15 & 2.3 & .01 \\ .01 & .01 & 1.00 \end{bmatrix}.$$

Si $\beta = 10$, $t = 2$ *et l'on arrondit, alors*

$$\hat{g}_{11} = 10, \qquad \hat{g}_{21} = 1.5, \qquad \hat{g}_{31} = 0.001, \qquad \hat{g}_{22} = 0.00.$$

L'algorithme s'arrête au calcul de $\hat{g}_{32}$.

Exercice 6.1

(a) Soient $\alpha \in \mathbf{R}$, v, $w \in \mathbf{R}^{n-1}$ et $B \in \mathbf{R}^{(n-1)\times(n-1)}$. Si

$$A = \begin{bmatrix} \alpha & w^T \\ v & B \end{bmatrix}$$

est définie positive, montrer que $B - vw^T/\alpha$ est aussi définie positive.

(b) Soit $A \in \mathbf{R}^{n\times n}$ symétrique et définie positive. Montrer que

$$|a_{ij}| \leq (a_{ii} + a_{jj})/2, \qquad i \neq j,$$

et

$$|a_{ij}| \leq \sqrt{a_{ii}a_{jj}}.$$

(c) Soit la matrice $A + iB \in \mathbf{C}^{n\times n}$ hermitienne et définie positive. Montrer que

$$C = \begin{bmatrix} A & -B \\ B & A \end{bmatrix}$$

est définie positive

(d) Monter l'unicité de la décomposition de Cholesky d'une matrice symétrique et définie positive.

(e) Montrer que la matrice suivante n'est pas définie positive:

$$A = \begin{bmatrix} 36 & 30 & 18 \\ 30 & 41 & 23 \\ 18 & 23 & 12 \end{bmatrix}.$$

6.3 Systèmes symétriques indéfinis

Chapitre 7

Systèmes surdéterminés

7.1 Le problème mathématique des moindres carrés

Soit $A \in \mathbf{R}^{m\times n}, m \geq n$, et $b \in \mathbf{R}^n$. Le système surdéterminé

$$Ax = b \tag{7.1}$$

n'admet pas de solution exacte si

$$b \notin R(A) := \text{lin}\{a_1, \ldots, a_n\} \subset \mathbf{R}^n.$$

Le problème des moindres carrés consiste à trouver un x tel que le mininum

$$\min_x \|Ax - b\|_2 \tag{7.2}$$

est atteint.

Trois procédés de résolution de ce problème viennent à l'esprit.

(i) On verra que (7.2) est équivalent à

$$A^T Ax = A^T b \tag{7.3}$$

qu'on appelle *équations normales*. On remarque que

$$A^T A \geq 0 \quad \text{et} \quad \text{rg}\, A^T A = \text{rg}\, A \leq n. \tag{7.4}$$

Si $\text{rg}\, A = n$, alors $A^T A > 0$, et on a la décompositon de Cholesky $A^T A = GG^T, G$ triangulaire inférieure. On résout (7.3) par substitutions avant puis arrière:

$$Gy = A^T b, \qquad G^T x = y.$$

(ii) Dans la norme 2, si Q est orthogonale: $Q \in \mathbf{R}^{m\times m}, Q^TQ = I_m$, alors

$$\|Q^TAx - Q^Tb\|_2 = \|Ax - b\|_2.$$

Si l'on choisit Q telle que

$$Q^TA = R$$

soit triangulaire supérieure, A se décompose selon Householder ou Givens:

$$A = QR.$$

Si $\operatorname{rg} A = n$, on peut résoudre pour x par substitution arrière:

$$Rx = Q^Tb.$$

(iii) Si $\operatorname{rg} A < n \leq m$, on emploie la décomposition QR avec pivotage sur les colonnes, ou la décomposition en valeurs singulières:

$$A = U\Sigma V^T, \qquad U\Sigma V^Tx = b, \qquad y = V^Tx, \qquad \Sigma y = U^Tb.$$

Tableau 7.1. Méthodes de résolution du probème des moindres carrés ($m \geq n$)

Algorithme	**Nombre de flops**
Equations normales	$\frac{mn^2}{2} + \frac{n^3}{6}$
Orthogonalisation de Householder	$mn^2 - \frac{n^3}{3}$
Gram–Schmidt modifié	mn^2
Orthogonalisation de Givens	$2mn^2 - \frac{2}{3}n^3$
Biorthogonalisation de Householder	$2mn^2 - \frac{2}{3}n^3$
R-biorthogonalisation	$mn^2 + n^3$
Décomposition en valeurs singulières de Golub-Reinsch	$2mn^2 + 4n^3$
Décomposition en valeurs singulières de Chan	$mn^2 + \frac{17}{3}n^3$

Théorème 7.1 *Soit l'ensemble minimisant des moindres carrés*

$$X = \{x \in \mathbf{R}^n;\ \|Ax - b\|_2 = \min\}, \tag{7.5}$$

alors

(1) $x \in X \iff A^T(b - Ax) = 0$ *(éq. normales);*

(2) X *est* convexe*, c'est-à-dire, si* $x_1, x_2 \in X$*, alors*

$$\alpha x_1 + (1-\alpha)x_2 \in X, \forall\alpha, 0 \leq \alpha \leq 1;$$

(3) *l'élément* $x_{\mathrm{mc}} \in X$ *tel que* $\|x_{\mathrm{mc}}\|_2 = \min$ *est* unique*;*

(4) $X = \{x_{\mathrm{mc}}\} \iff \operatorname{rg} A = n$.

Démonstration. On démontre les énoncés (1) à (4).

L'énoncé (1) suit facilement de la démonstration du prochain théorème 7.2 qui

emploie la décomposition de A selon les valeurs singulières, comme on dira au corollaire 7.1 qui suit le théorème 7.2. On a l'habitude de montrer que $x \in X \Rightarrow A^T(b - Ax) = 0$ en prenant le gradient de la fonction

$$f(x) = \|Ax - b\|_2^2 = [Ax - b]^T[Ax - b] = x^T A^T Ax - 2x^T A^T b + b^T b.$$

Si l'on pose

$$S = A^T A = (s_{ij}) = (s_{ji}),$$

alors

$$f(x) = \sum_{i,j=1}^{n} s_{ij} x_i x_j - 2 \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} b_i x_j + \text{const.}$$

Du fait que $s_{ij} = s_{ji}$, la $k^{\text{ième}}$ composante du gradient de f au point x est

$$\frac{\partial f}{\partial x_k} = 2 \sum_{j=1}^{n} s_{kj} x_j - 2 \sum_{i=1}^{m} a_{ik} b_i.$$

Puisque le gradient s'annule à un minimum, tout x qui minimise f satisfait les équations

$$\sum_{j=1}^{n} s_{kj} x_j = \sum_{i=1}^{m} a_{ik} b_i, \qquad k = 1, 2, \ldots, n,$$

c'est-à-dire

$$A^T(Ax - b) = 0.$$

Si $\operatorname{rg} A = n$, la réciproque, $\Leftarrow$, suit facilement du fait que

$$Q(x) = \|Ax - b\|_2^2$$

est une forme quadratique non dégénérée; alors x minimise $Q(x)$ ssi

$$\partial_x Q(x) = 2A^T(b - Ax) = 0.$$

Pour (2) on vérifie que X est convexe. Soient $x_1, x_2 \in X$ et $0 \leq \alpha \leq 1$. Alors

$$\begin{aligned} \|A[\alpha x_1 + (1-\alpha)x_2] - b\|_2 &\leq \alpha \|Ax_1 - b\|_2 + (1-\alpha)\|Ax_2 - b\|_2 \\ &= \min \|Ax - b\|_2; \end{aligned}$$

il s'ensuit que

$$\alpha x_1 + (1 - \alpha)x_2 \in X.$$

Pour (3), on sait que $0 \notin X$ et que X est convexe. Donc le point $x_{\text{mc}} \in X$ le plus près de l'origine est unique; sinon les points d'un segment de droite serait tous à distance $\|x_{\text{mc}}\|$ de l'origine, ce qui est impossible parce que le segment de droite serait alors un arc de cercle.

Pour (4) on montre d'abord l'implication $\Leftarrow$. On sait que

$$\operatorname{rg} A = n \Longleftrightarrow \operatorname{rg} A^T A = n;$$

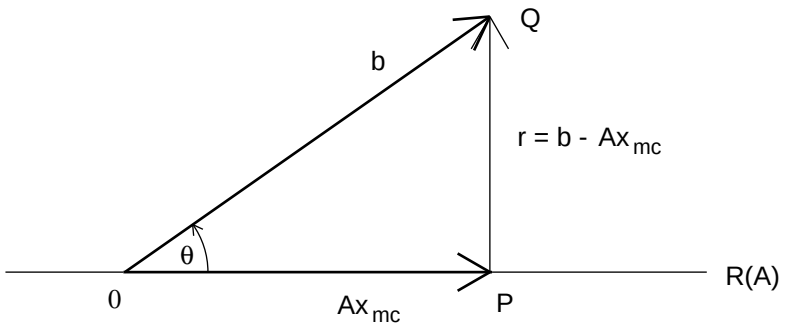

Figure 7.1: Le vecteur $\vec{OP} = Ax_{\rm mc}$ est la projection orthogonale de $b = \vec{OQ}$ sur $R(A)$. Le point P est l'élément de $R(A)$ le plus près du point Q. L'angle θ est donné par la relation $\sin\theta = (\rho_{\rm mc})/\|b\|_2$ où $\rho_{\rm mc} = \|r\|_2 = \|b - Ax_{\rm mc}\|_2$.

alors la solution de

$$A^T Ax = a^T b$$

est unique.
D'autre part $x_{\rm mc}$ unique $\Rightarrow$ la solution de $A^T Ax = a^T b$ est unique $\Rightarrow \operatorname{rg} A^T A = n$. □

Définition 7.1 *On appelle respectivement le* résidu *la différence*

$$r = b - Ax, \tag{7.6}$$

les équations normales *le système symétrique*

$$A^T Ax = A^T b \tag{7.7}$$

et la somme des moindres carrés *la norme*

$$\rho_{\rm mc}^2 = \|Ax_{\rm mc} - b\|_2^2. \tag{7.8}$$

La décomposition en valeurs singulières donne une vue pénétrante du problème des moindres carrés.

Théorème 7.2 *Soient $A \in \mathbf{R}^{m\times n}$, $m \geq n$, et $r = \operatorname{rg} A$. Considérons la décomposition en valeurs singulières de A:*

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U\Sigma V^T$$

donnée par les matrices orthogonales

$$U = [u_1, \ldots, u_m] \in \mathbf{R}^{m\times m}, \quad V = [v_1, \ldots, v_n] \in \mathbf{R}^{n\times n}.$$

Soit

$$\|Ax_{\rm mc} - b\|_2 = \rho_{\rm mc} = \min.$$

Alors

$$x_{\rm mc} = \sum_{i=1}^{r} \left(\frac{u_i^T b}{\sigma_i} \right) v_i \tag{7.9}$$

et

$$\rho_{\rm mc}^2 = \sum_{i=r+1}^{m} \left(u_i^T b \right)^2 . \tag{7.10}$$

Démonstration. Soit $x \in \mathbf{R}^n$ et posons $\alpha = V^T x$; alors

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\| U^T A V V^T x - U^T b \right\|_2^2 \\ &= \left\| \Sigma V^T x - U^T b \right\|_2^2 \\ &= \sum_{i=1}^{r} \left(\sigma_i \alpha_i - u_i^T b \right)^2 + \sum_{i=r+1}^{m} \left(u_i^T b \right)^2 . \end{aligned}$$

On minimise le 2^e membre en prenant

$$\alpha_i = \frac{u_i^T b}{\sigma_i};$$

alors

$$\rho_{\rm mc}^2 = \sum_{i=r+1}^{m} \left(u_i^T b \right)^2 .$$

La solution $x = V\alpha$ satisfait $\|x\|_2^2 = \min$ si l'on prend $\alpha_{r+1} = \alpha_{r+2} = \ldots = \alpha_n = 0$; en effet $\|\alpha\|_2 = \|V^T x\|_2 = \|x\|_2$ parce que V est orthogonale. □

Corollaire 7.1 *On démontre la partie* (1) *du théorème 7.1 au moyen de la décomposition de $A^T A$ en valeurs singulières.*

Démonstration. La démonstration découle des implications suivantes:

$$A^T A x - A^T b = 0,$$

$$V \Sigma^T U^T U \Sigma V^T x - V \Sigma^T U^T b = 0,$$

$$\Sigma^T \Sigma V^T x - \Sigma^T U^T b = 0, \qquad \left(\text{on a appliqué } V^{-1} \text{ à gauche} \right),$$

$$\Sigma^T \Sigma \alpha - \Sigma^T U^T b = 0, \qquad \left(\alpha = V^T x \right),$$

$$\sigma_i \alpha_i - u_i^T b = 0, \qquad i = 1, \ldots, r,$$

$$\|Ax - b\|_2 = \min .$$

On peut aussi remonter de la dernière à la première relation. □

On récapitule les résultats précédents au moyen de l'*inverse généralisée* ou *pseudo-inverse* de A, notée A^+.

Soient $A \in \mathbf{R}^{m\times n}, m \geq n$, de rang r et les matrices orthogonales

$$U = [u_1, \ldots, u_m] \in \mathbf{R}^{m\times m}, \quad V = [v_1, \ldots, v_n] \in \mathbf{R}^{n\times n}$$

qui donnent la décomposition en valeurs singulières de A:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U\Sigma V^T,$$

où $\Sigma \in \mathbf{R}^{m\times n}$.

Définition 7.2 *L'inverse généralisée de A est la matrice $A^+ \in \mathbf{R}^{n\times m}$ définie par l'expression:*

$$A^+ = V\Sigma^+ U^T, \tag{7.11}$$

où

$$\Sigma^+ = \operatorname{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right) \in \mathbf{R}^{n\times m}. \tag{7.12}$$

Théorème 7.3 *Soit A^+ l'inverse généralisé de A et $\operatorname{rg} A = r$. Alors la solution du problème des moindres carrés est*

$$x_{\mathrm{mc}} = A^+ b \tag{7.13}$$

et

$$\rho_{\mathrm{mc}} = \left\| \left(I - AA^+\right) b \right\|_2 . \tag{7.14}$$

Démonstration. Par le théorème précédent, on a

$$A^+ b = V\Sigma^+ U^T b = \sum_{i=1}^{r} \left(\frac{u_i^T b}{\sigma_i}\right) v_i = x_{\mathrm{mc}}$$

De plus,

$$\begin{aligned}
I - AA^+ &= I_m - U\Sigma V^T V\Sigma^+ U^T \\
&= I_m - U\Sigma\Sigma^+ U^T \\
&= \begin{bmatrix} I_r & 0 \\ 0 & I_{m-r} \end{bmatrix} - U \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0_{m-r} \end{bmatrix} \begin{bmatrix} 1/\sigma_1 & 0 & \ldots & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & \cdots & 1/\sigma_r & 0 \\ 0 & \cdots & 0 & 0_{m-r} \end{bmatrix} U^T \\
&= \begin{bmatrix} I_r & 0 \\ 0 & I_{m-r} \end{bmatrix} - U \begin{bmatrix} I_r & 0 \\ 0 & 0_{m-r} \end{bmatrix} U^T \\
&= U \left\{ \begin{bmatrix} I_r & 0 \\ 0 & I_{m-r} \end{bmatrix} - \begin{bmatrix} I_r & 0 \\ 0 & 0_{m-r} \end{bmatrix} \right\} U^T \\
&= U \begin{bmatrix} 0 & 0 \\ 0 & I_{m-r} \end{bmatrix} U^T.
\end{aligned}$$

Alors

$$\begin{aligned}\left(I - AA^+\right) b &= U \begin{bmatrix} 0 & 0 \\ 0 & I_{m-r} \end{bmatrix} U^T b \\ &= U \begin{bmatrix} 0 & 0 \\ 0 & I_{m-r} \end{bmatrix} \begin{bmatrix} u_1^T b \\ u_2^T b \\ \vdots \\ u_m^T b \end{bmatrix} = U \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{r+1}^T b \\ \vdots \\ u_m^T b \end{bmatrix};\end{aligned}$$

donc

$$\begin{aligned}\left\|(I - AA^+)b\right\|_2 &= \left\| U \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{r+1}^T b \\ \vdots \\ u_m^T b \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{r+1}^T b \\ \vdots \\ u_m^T b \end{bmatrix} \right\|_2^2 \\ &= \sum_{i=r+1}^{m} \left(u_i^T b\right)^2 = \rho_{\text{mc}}^2. \quad \square\end{aligned}$$

On considère maintenant la question de la stabilité du problème des moindres carrés. On énonce, sans démonstration, un théorème sur la perturbation de A^+.

Théorème 7.4 *Soient $A, \delta A \in \mathbf{R}^{m\times n}$. Alors*

$$\left\|(A+\delta A)^+ - A^+\right\|_F \leq 2\|\delta A\|_F \max\left\{ \left\|A^+\right\|_2^2, \left\|(A+\delta A)^+\right\|_2^2 \right\}. \qquad (7.15)$$

Démonstration. Voir Weddon ou Stewart.

Remarque 7.1 L'inégalité (7.15) généralise l'inégalité de Banach pour A régulière:

$$\left\|(A+\delta A)^{-1} - A^{-1}\right\| \leq \|\delta A\| \ \left\|A^{-1}\right\| \ \left\|(A+\delta A)^{-1}\right\|.$$

Mais le second membre de (7.15) ne tend pas nécessairement vers zéro quand $\|\delta A\|_F \to 0$.

Exemple 7.1 *Soient*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \delta A = \begin{bmatrix} 0 & 0 \\ 0 & \epsilon \\ 0 & 0 \end{bmatrix};$$

alors

$$A = \Sigma, \qquad A^+ = V\Sigma^+ U^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A + \delta A = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \\ 0 & 0 \end{bmatrix} = \Sigma_{(A+\delta A)}, \qquad (A+\delta A)^+ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\epsilon & 0 \end{bmatrix},$$

$$\left\|(A+\delta A)^+ - A^+\right\|_2 = \left\| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/\epsilon & 0 \end{bmatrix} \right\|_2 = 1/\epsilon.$$

Pour simplifier le théorème suivant, on suppose dans la suite que A et $A+\delta A$ sont de rang plein:

$$\operatorname{rg} A = \operatorname{rg}(A+\delta A) = n \ (\leq m).$$

On généralise la notion de conditionnement d'une matrice:

$$\kappa(A) = \|A\| \, \|A^+\|, \qquad A \in \mathbf{R}^{m\times n}. \tag{7.16}$$

Dans le cas particulier de la norme 2,

$$\kappa_2(A) = \sigma_1(A)/\sigma_r(A), \qquad r = \operatorname{rg}(A).$$

On présente maintenant un second théorème de stabilité sur la solution du problème des moindres carrés.

Théorème 7.5 *Si x, r, $\hat{x}$ et $\hat{r}$ satisfont*

$$\|Ax - b\|_2 = \min, \qquad r = b - Ax,$$

$$\|(A+\delta A)\hat{x} - (b+\delta b)\|_2 = \min, \qquad \hat{r} = (b+\delta b) - (A+\delta A)\hat{x},$$

où

$$A, \delta A \in \mathbf{R}^{m\times n}, m \geq n, b, \delta b \in \mathbf{R}^m, b \neq 0;$$

si, de plus,

$$\epsilon = \max\left\{ \frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta b\|_2}{\|b\|_2} \right\} \leq \frac{\sigma_n(A)}{\sigma_1(A)} = \frac{1}{\kappa_2(A)}$$

et

$$\sin\theta = \frac{\rho_{\mathrm{mc}}}{\|b\|_2} \neq 1;$$

alors

$$\frac{\|\hat{x} - x\|_2}{\|x\|_2} \leq \epsilon \left\{ \frac{2\kappa_2(A)}{\cos\theta} + \kappa_2(A)^2 \tan\theta \right\} + O\left(\epsilon^2\right), \tag{7.17}$$

et

$$\frac{\|\hat{r} - r\|_2}{\|b\|_2} \leq \epsilon[1 + 2\kappa_2(A)] \min\{1, m-n\} + O\left(\epsilon^2\right). \tag{7.18}$$

Démonstration. Voir Golub–Van Loan.

Remarque 7.2 Le second membre de (7.17) contient le terme $\kappa_2(A)^2 \tan\theta$; si $\theta \neq 0$, la stabilité de $x_{\rm mc}$ dépend du carré de $\kappa_2(A)$.
Mais la stabilité de r ne dépend que de la $1^{\text{ière}}$ puissance de $\kappa_2(A)$.

Exemple 7.2 *Soient $\beta = 10$, $t = 4$ et considérons*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-6} \\ 0 & 0 \end{bmatrix}, \quad \delta A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 10^{-8} \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \delta b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

On voit que

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} 1 \\ 0.999\,9 \times 10^{4} \end{bmatrix}, \quad r = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad \hat{r} \approx \begin{bmatrix} 0 \\ 0.999\,9 \times 10^{-2} \\ -0.999\,9 \times 10^{0} \end{bmatrix}.$$

Puisque $\kappa_2(A) = 10^6$ et

$$\sin\theta = \frac{\rho_{\rm mc}}{\|b\|_2} = \frac{1}{\sqrt{2}} \Rightarrow \theta = \frac{\pi}{4} \Rightarrow \tan\theta = 1,$$

il s'ensuit que

$$\frac{\|\hat{x} - x\|_2}{\|x\|_2} \approx 0.999\,9 \times 10^4 \leq \kappa_2^2(A) \frac{\|\delta(A)\|_2}{\|A\|_2} = 10^{12} \times 10^{-8} = 10^4$$

et

$$\frac{\|\hat{r} - r\|_2}{\|b\|_2} \approx 0.707 \times 10^{-2} \leq 2\kappa_2(A) \frac{\|\delta(A)\|_2}{\|A\|_2} = 2 \times 10^6 \times 10^{-8} = 2 \times 10^{-2}.$$

On donne les calculs en détail.

$$A^T A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-12} \end{bmatrix}, \quad A^T b = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 10^{-12} \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \Rightarrow \quad x = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$(A+\delta A)^T (A+\delta A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^{-6} & 10^{-8} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 10^{-6} \\ 0 & 10^{-8} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-12} + 10^{-16} \end{bmatrix},$$

$$(A + \delta A)^T (b + 0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^{-6} & 10^{-8} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 10^{-8} \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 10^{-12} + 10^{-16} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 10^{-8} \end{bmatrix} \Rightarrow \hat{x} \approx \begin{bmatrix} 1 \\ 0.999\,9 \times 10^4 \end{bmatrix},$$

$$\frac{10^{-8}}{10^{-12}(1 + 10^{-4})} = \frac{10^4}{1 + 0.000\,1} = 0.000\,900\,010\,00 \times 10^4 \approx 0.999\,9 \times 10^4.$$

Au $2^{\text{ième}}$ *dénominateur on fait* $1 + 0.000\,1 = 1.000\,1$ *avec* $t = 5$ *!!!*

$$r = b - Ax = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 10^{-6} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$\begin{aligned} \hat{r} &= b - (A + \delta A)\hat{x} \approx \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 10^{-6} \\ 0 & 10^{-8} \end{bmatrix} \begin{bmatrix} 1 \\ 0.999\,9 \times 10^{4} \end{bmatrix} \\ &\approx \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0.999\,9 \times 10^{-2} \\ 0.999\,9 \times 10^{-4} \end{bmatrix} \approx \begin{bmatrix} 1 \\ -0.999\,9 \times 10^{-2} \\ 0.999\,9 \times 10^{0} \end{bmatrix}. \end{aligned}$$

Ici $1.000\,0 - 0.000\,099\,99 = 0.999\,900\,01 \approx 0.999\,9$.

$$A^{+} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^{6} & 0 \end{bmatrix}, \quad \kappa_2(A) = \|A\|_2 \, \|A^{+}\|_2 = 1 \times 10^{6} = 10^{6}.$$

$$\|\hat{x} - x\|_2 = \left\| \begin{bmatrix} 0 \\ 0.999\,9 \times 10^{4} \end{bmatrix} \right\|_2 = 0.999\,9 \times 10^{4}, \quad \|x\|_2 = \left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_2 = 1$$

$$\Rightarrow \frac{\|\hat{x} - x\|_2}{\|x\|_2} \approx 0.999\,9 \times 10^{4}.$$

$$\kappa_2(A)^2 \frac{\|\delta a\|_2}{\|A\|_2} = 10^{12} \times \frac{10^{-8}}{1} = 10^{4},$$

$$\|\hat{r} - r\| = \left\| \begin{bmatrix} 0 \\ -0.999\,9 \times 10^{-2} \\ 0.000\,1 \end{bmatrix} \right\|_2 \approx \sqrt{10^{-4} + 10^{-4}} \approx 0.707\,0 \times 10^{-2}.$$

On applique le théorème précédent à l'équation normale $A^T Ax = A^T b$.

Algorithme 7.1 *Soit* $A \in \mathbf{R}^{m \times n}$, $\operatorname{rg}(A) = n \leq m$ *et* $b \in \mathbf{R}^m$. *L'algorithme calcule* x_{mc} *tel que* $\|Ax_{\text{mc}} - b\|_2 = \min$.

$C := A^T A$
$d := A^T b$
$C = GG^T$ *(Décomposition de Cholesky)*
Résoudre:
$Gy = d$ *(Algorithme 4.1-1)*
et
$G^T x = y$ *(Algorithme 4.1-2)* □

L'algorithme requiert $\frac{n^2}{2}(m + \frac{n}{3})$ flops.

Remarque 7.3 $A^T A > 0$ ssi $\operatorname{rg} A = n$.

Remarque 7.4 Si C et d sont sans erreur,

$$(A^TA + E)\hat{x} = A^Tb, \qquad \|E\|_2 \approx \mathbf{u}\|A^TA\|_2$$

$$\Rightarrow \frac{\|\hat{x} - x_{\rm mc}\|_2}{\|x_{mc}\|_2} \approx \mathbf{u}\kappa_2(A^TA) = \mathbf{u}\kappa_2(A)^2,$$

en accord avec le théorème précédent. Mais si

$$\rho_{\rm mc} = \|Ax_{\rm mc} - b\|_2 \approx 0,$$

alors la méthode des équations normales peut introduire une erreur plus forte que celle escomptée.

Exemple 7.3 *Soient*

$$A = \begin{bmatrix} 1 & 1 \\ 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ 10^{-3} \\ 10^{-3} \end{bmatrix}.$$

En arithmétique exacte les équations normales $A^TAx = A^Tb := d$ donnent

$$\begin{bmatrix} 1+10^{-6} & 1 \\ 1 & 1+10^{-6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2+10^{-6} \\ 2+10^{-6} \end{bmatrix} \qquad \Rightarrow \qquad x_{\rm mc} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

et $\rho_{\rm mc} = 0$.
Avec $\beta = 10$, $t = 6$ et calculs tronqués,

$$A^TA = \begin{bmatrix} 1+10^{-6} & 1 \\ 1 & 1+10^{-6} \end{bmatrix} \approx \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

qui est singulière du fait que $1.000\,001 \approx 1.000\,00$.
Mais avec $\beta = 10$, $t = 7$ et calculs tronqués, on a

$$\begin{bmatrix} 1.000\,001 & 1 \\ 1 & 1.000\,001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.000\,001 \\ 2.000\,001 \end{bmatrix} \quad \Rightarrow \quad \hat{x} = \begin{bmatrix} 2.000\,001 \\ 0 \end{bmatrix}.$$

On obtient le même résultat par la décomposition de Cholesky avec $t = 7$:

$$g_{11} = \sqrt{1.000\,001} \approx 1.000\,000$$

$$g_{21} = a_{21}/g_{11} = 1$$

$$g_{22} = \sqrt{a_{22} - g_{21}^2} = \sqrt{10^{-6}} = 10^{-3},$$

ce qui donne pour $Gy = d$ et $G^Tx = y$:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2.000\,001 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} 2.000\,001 \\ 0 \end{bmatrix}.$$

Or

$$A = U \begin{bmatrix} 1.4142 & 0 \\ 0 & 0.0010 \\ 0 & 0 \end{bmatrix} V^T \qquad \Rightarrow \qquad \|A\|_2 = 1.4142,$$

$$A^+ = V \begin{bmatrix} 1/1.44142 & 0 & 0 \\ 0 & 10^3 & 0 \end{bmatrix} U \qquad \Rightarrow \qquad \|A^+\|_2 = 10^3.$$

Alors

$$\kappa_2(A) = 1.4142 \times 10^3.$$

Enfin

$$\frac{\|\hat{x} - x_{\mathrm{mc}}\|_2}{\|x_{\mathrm{mc}}\|_2} = \frac{\sqrt{1.000\,001^2 + 1^2}}{\sqrt{2}} \approx 1.000\,000 \approx \mathbf{u}\kappa_2(A)^2$$

où

$$\mathbf{u}\kappa_2(A)^2 \approx 10^{1-7} \times 2 \times 10^6 = 2 \times 10^0.$$

Ce résultat numérique illustre la remarque précédente.

Exercice 7.1

(a) Soit $x \in X$, où X est l'ensemble minimisant de $\|Ax - b\|$. Montrer au moyen de l'identité

$$\|A(x + \alpha w) - b\|_2^2 = \|Ax - b\|_2^2 + 2\alpha w^T A^T (Ax - b) + \alpha^2 \|Aw\|_2^2,$$

que

$$A^T A x = A^T b.$$

(b) Soit la fonction $\phi : \mathbf{R}^n \to \mathbf{R}$ définie par

$$\phi(x) = \frac{1}{2}\|Ax - b\|_2^2.$$

Montrer que le gradient $\nabla\phi(x)$ est

$$\nabla\phi(x) = A^T(Ax - b).$$

(c) Soit $A \in \mathbf{R}^{m \times n}$ et $\operatorname{rg} A = n$. Si $\|E\|_2 \|A^+\|_2 < 1$, montrer que

$$\operatorname{rg}(A + E) = n.$$

(d) Soient

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \qquad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Résoudre le problème des moindres carrés pour $Ax = b$ au moyen des équations normales.

7.2 Orthogonalisations de Householder et de Gram–Schmidt

Soient $A \in \mathbf{R}^{m\times n}$, $m \geq n$ et $b \in \mathbf{R}^n$. Si l'on a calculé la matrice orthogonale $Q \in \mathbf{R}^{m\times m}$ de la décomposition QR de A:

$$Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \quad \begin{matrix} n \\ m-n \end{matrix}$$

où R est triangulaire supérieure, et si

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix} \quad \begin{matrix} n \\ m-n, \end{matrix}$$

alors

$$\|Ax-b\|_2^2 = \left\|Q^T Ax - Q^T b\right\|_2^2 = \|R_1 x - c\|_2^2 + \|d\|_2^2 \quad \forall x \in \mathbf{R}^n.$$

Si

$$\mathrm{rg}(A) = \mathrm{rg}(R_1) = n,$$

le vecteur minimisant x_{mc} est solution du système

$$R_1 x_{\mathrm{mc}} = c$$

et le résidu est

$$\rho_{\mathrm{mc}} = \|d\|_2.$$

Conclusion. Le problème des moindres carrés de rang plein se résout facilement si l'on connaît la *décomposition* QR de A:

$$A = QR.$$

On obtient la décomposition QR au moyen de réflexions de Householder. Il suffit de considérer un petit exemple: $m = 6, n = 5$. Supposons qu'on ait déjà calculé les matrices de Householder P_1 et P_2 telles que

$$P_2 P_1 A = \begin{bmatrix} \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ 0 & \mathrm{x} & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ 0 & 0 & \mathrm{y} & \mathrm{x} & \mathrm{x} \\ 0 & 0 & \mathrm{y} & \mathrm{x} & \mathrm{x} \\ 0 & 0 & \mathrm{y} & \mathrm{x} & \mathrm{x} \\ 0 & 0 & \mathrm{y} & \mathrm{x} & \mathrm{x} \end{bmatrix}.$$

On détermine la matrice de Householder $\tilde{P}_3 \in \mathbf{R}^{4\times 4}$ telle que

$$\tilde{P}_3 \begin{bmatrix} y \\ y \\ y \\ y \end{bmatrix} = \begin{bmatrix} x \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

et on pose $P_3 = \text{diag}\left(I_2, \tilde{P}_3\right)$. Alors,

$$P_3P_2P_1A = \begin{bmatrix} \text{x} & \text{x} & \text{x} & \text{x} & \text{x} \\ 0 & \text{x} & \text{x} & \text{x} & \text{x} \\ 0 & 0 & \text{x} & \text{x} & \text{x} \\ 0 & 0 & 0 & \text{x} & \text{x} \\ 0 & 0 & 0 & \text{x} & \text{x} \\ 0 & 0 & 0 & \text{x} & \text{x} \end{bmatrix}.$$

Dans le cas général, $P_nP_{n-1}\cdots P_1A = R$ est triangulaire supérieure; on obtient $A = QR$ en posant $Q = P_1\cdots P_n$.

Algorithme 7.2 (Orthogonalisation de Householder) *Soient $A \in \mathbf{R}^{m\times n}, m \geq n$. L'algorithme produit la matrice orthogonale Q telle que $Q^TA = R$ soit triangulaire supérieure; R renouvelle A.*

Pour $k = 1, \ldots, n$
  Déterminer la matrice de Householder $\tilde{P}_k$ d'ordre $m-k+1$ telle que
    $\tilde{P}_k(a_{kk}, \ldots, a_{mk})^T = (r_{kk}, 0, \ldots, 0)^T$
  $A := \text{diag}\left(I_{k-1}, \tilde{P}_k\right) A$ □

L'algorithme requiert $n^2(m - n/3)$ flops.

Remarque 7.5 On conserve Q en facteurs; toute l'information associée à

$$P_k = \text{diag}\left(I_{k-1}, \tilde{P}_k\right) := I_m - \beta_k v^{(k)}\left[v^{(k)}\right]^T$$

se retrouve sous la forme

$$A := \begin{bmatrix} v_1^{(1)} & r_{12} & r_{13} & r_{14} & r_{15} \\ v_2^{(1)} & v_2^{(2)} & r_{23} & r_{24} & r_{25} \\ v_3^{(1)} & v_3^{(2)} & v_3^{(3)} & r_{34} & r_{35} \\ v_4^{(1)} & v_4^{(2)} & v_4^{(3)} & v_4^{(4)} & r_{45} \\ v_5^{(1)} & v_5^{(2)} & v_5^{(3)} & v_5^{(4)} & v_5^{(5)} \\ v_6^{(1)} & v_6^{(2)} & v_6^{(3)} & v_6^{(4)} & v_6^{(5)} \end{bmatrix}, \quad \begin{bmatrix} r_{11} \\ r_{22} \\ r_{33} \\ r_{44} \\ r_{55} \end{bmatrix}, \quad \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix}.$$

Si l'on doit accumuler $Q = P_1P_2\cdots P_n$, on fait

$Q := I$
Pour $k = n, \ldots, 1$
  $Q := P_kQ$

L'accumulation requiert $2\left[m^2n - mn^2 + n^3/3\right]$ flops.

Pour résoudre le problème des moindres carrés, il suffit d'appliquer les P_k sur b, sans former Q, en $O(nm)$ flops:

Pour $k = 1, \ldots, n$

$$Q := P_k b$$

Ensuite pour obtenir x on fait la substitution arrière:

$x_n := b_n / r_n$
Pour $i = n-1, \ldots, 1$
$\quad x_i := \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \Big/ r_i$

On a supposé que R est de la forme donnée plus haut. La solution du problème des moindres carrés de rang plein requiert en tout $n^2(m - n/3)$ flops. On peut démontrer que la solution $\hat{x}$ satisfait:

$$\|(A + \delta A)\hat{x} - (b + \delta b)\|_2 = \min,$$

où

$$\|\delta A\|_2 \leq (6m - 3n + 41)n\mathbf{u}\|A\|_F + O\left(\mathbf{u}^2\right)$$

et

$$\|\delta b\|_2 \leq (6m - 3n + 40)n\mathbf{u}\|b\|_2 + O\left(\mathbf{u}^2\right).$$

Remarque 7.6 La forte stabilité de Householder exige le double du nombre de flops de la méthode des équations normales.

Remarque 7.7 La méthode de Householder pour le problème des moindres carrés s'arrête si $\operatorname{rg} A < n \leq m$. On aura des difficultés si

$$\kappa_2(A) \approx 1/\mathbf{u};$$

mais c'est pire avec les équations normales où la limite est

$$\kappa_2(A) \approx 1/\sqrt{\mathbf{u}}.$$

Exemple 7.4 *Soient*

$$A = \begin{bmatrix} 1 & 1 \\ 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ 10^{-3} \\ 10^{-3} \end{bmatrix}.$$

Avec $\beta = 10, t = 6$ et l'arithmétique tronquée, Householder donne

$$\hat{x}_{\text{mc}} = \begin{bmatrix} 1.000\,00 \\ 1.000\,00 \end{bmatrix}.$$

D'autre part, la matrice des équations normales est singulière:

$$A^T A \approx \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

comme on a vu à la section précédente.

Problème de la base orthonormée. Soient n vecteurs indépendants:

$$a_1, \ldots, a_n \in \mathbf{R}^m.$$

Trouver une *base orthonormée* de l'enveloppe linéaire de ces vecteurs.

Résolution. On emploie l'algorithme 7.2. En fait, si

$$A = [a_1, \ldots, a_n] = QR, \qquad Q = [q_1, \ldots, q_m],$$

alors, pour $k = 1, \ldots, n$,

$$a_k = \sum_{i=1}^{k} r_{ik} q_i,$$

et par l'orthogonalité de Q:

$$r_{ik} = q_i^T a_k,$$

c'est-à-dire

$$\operatorname{lin}\{a_1, \ldots, a_k\} = \operatorname{lin}\{q_1, \ldots, q_k\}, \qquad k = 1, 2, \ldots, n;$$

en particulier,

$$R(A) = \operatorname{lin}\{q_1, \ldots, q_n\}. \quad \square$$

De plus la méthode de Householder produit une base orthonormée pour le supplémentaire orthogonal de l'image de A:

$$R(A)^T = \operatorname{lin}\{q_{n+1}, \ldots, q_m\}.$$

Considérons maintenant les algorithmes de Gram–Schmidt pour obtenir une base orthogonale.

L'**algorithme de Gram–Schmidt classique** (GSC). On suppose que $\operatorname{rg} A = n$; alors $r_{kk} \neq 0, k = 1, \ldots, n$; on peut donc obtenir

$$q_k = \frac{1}{r_{kk}} \left(a_k - \sum_{i=1}^{k-1} r_{ik} q_i \right).$$

On voit que q_k est un vecteur unité parallèle à

$$z_k = a_k - \sum_{i=1}^{k-1} s_{ik} q_i;$$

pour que z_k soit orthogonal à $q_1, \ldots, q_{k-1}$, c'est-à-dire

$$z_k \in [\operatorname{lin}\{q_1, \ldots, q_{k-1}\}]^{\perp},$$

on choisit

$$s_{ik} = q_i^T a_k, \qquad i = 1, \ldots, k-1.$$

Autrement dit, on fait

$$\begin{array}{ll} p_1 = a_1, & q_1 = \frac{p_1}{\|p_1\|}, \\ p_2 = a_2 - \mathrm{Proj}_{q_1} a_2, & q_2 = \frac{p_2}{\|p_2\|}, \\ p_3 = a_3 - \mathrm{Proj}_{q_1} a_3 - \mathrm{Proj}_{q_2} a_3, & q_3 = \frac{p_3}{\|p_3\|}, \\ \cdots & \\ p_n = a_n - \mathrm{Proj}_{q_1} a_n - \cdots - \mathrm{Proj}_{q_{n-1}} a_n, & q_n = \frac{p_n}{\|p_n\|}. \end{array}$$

Algorithme 7.3 (Algorithme GSC) *L'algorithme produit les colonnes q_k et r_k resp. de Q et R:*

Pour $k = 1, \ldots, n$
$s_{ik} := q_i^T a_k \qquad (i = 1, \ldots, k-1)$
$z_k := a_k - \sum_{i=1}^{k-1} s_{ik} q_i \quad \left(= a_k - \sum_{i=1}^{k-1} \mathrm{Proj}_{q_i} a_k\right)$
$r_{kk} := \left(z_k^T z_k\right)^{1/2} \quad (= \|z_k\|_2)$
$q_k := z_k / r_{kk} \quad (= z_k / \|z_k\|_2)$
$r_{ik} := s_{ik} / r_{kk}, \qquad (i = 1, \ldots, k-1)$ □

Remarque 7.8 La méthode GSC ne produit pas une base orthonormée pour $R(A)^T$ mais elle résout le problème des moindres carrés par la décomposition:

$$A = Q_1 R_1,$$

où $Q_1 \in \mathbf{R}^{m \times n}$, $R_1 \in \mathbf{R}^{n \times n}$, $Q_1^T Q_1 = I_n$ et R_1 est triangulaire supérieure.
En effet, de

$$(A^T A)x = A^T b,$$

il vient

$$R_1^T Q_1^T Q_1 R_1 x = R_1^T Q_1^T b \Rightarrow R_1^T R_1 x = R_1^T Q_1^T b \Rightarrow R_1 x = Q_1^T b.$$

Remarque 7.9 L'algorithme GSC est très pauvre parce qu'il cause une perte importante de l'orthogonalité de la base $\{q_1, q_2, \ldots\}$. A la $k^{\text{ième}}$ étape, GSC produit la $k^{\text{ième}}$ colonne de Q et la $k^{\text{ième}}$ colonne de R, q_k et r_k.

Algorithme de Gram–Schmidt modifié. On peut stabiliser le GSC par un simple réarrangement des calculs; on obtient alors l'algorithme de Gram–Schmidt modifié (GSM), qu'on peut décrire de la façon suivante: à la $k^{\text{ième}}$ étape on rend les vecteurs restants, $a_k, \ldots, a_n$, orthogonaux à q_{k-1}, c'est-à-dire

$$\begin{array}{lll} p_1 = a_1, & & q_1 = \|p_1\|^{-1} p_1, \\ p_k = a_k - \mathrm{Proj}_{q_1} a_k, & k = 2, 3, \ldots, n, & q_2 = \|p_2\|^{-1} p_2, \\ p_k = a_k - \mathrm{Proj}_{q_2} a_k, & k = 3, \ldots, n, & q_3 = \|p_3\|^{-1} p_3, \\ \cdots & & \\ p_k = a_k - \mathrm{Proj}_{q_{n-1}} a_k, & k = n, & q_n = \|p_n\|^{-1} p_n. \end{array}$$

Autrement dit à la $k^{\text{ième}}$ étape, GSM produit la $k^{\text{ième}}$ colonne de Q et la $k^{\text{ième}}$ ligne de R, q_k et r_k^T.

Pour dériver le GSM, on définit la matrice $A^{(k)}$:

$$\left[0, A^{(k)}\right] := A - \sum_{i=1}^{k-1} q_i r_i^T = \sum_{i=k}^{n} q_i r_i^T,$$

où les matrices partielles 0 et $A^{(k)}$ ont resp. $k-1$ et $n-k+1$ colonnes. Si

$$A^{(k)} = [z, B], \qquad z \in \mathbf{R}^m, B \in \mathbf{R}^{m\times(n-k)},$$

alors

$$r_{kk} = \|z\|_2$$

et

$$q_k = z/r_{kk}$$
$$(r_{k,k+1}, \ldots, r_{kn}) = q_k^T B.$$

Au pas suivant,

$$A^{(k+1)} = B - q_k(r_{k,k+1}, \ldots, r_{kn}). \quad \square$$

Algorithme 7.4 (Gram–Schmidt modifié) *Soit $A \in \mathbf{R}^{m\times n}$ et $\operatorname{rg} A = n$. L'algorithme décompose $A = Q_1 R_1$; les colonnes de $Q_1 \in \mathbf{R}^{m\times n}$ sont orthogonales et $R_1 \in \mathbf{R}^{n\times n}$ est triangulaire supérieure. Q_1 renouvelle A.*

Pour $k = 1, \ldots, n$
  $r_{kk} := \left(\sum_{i=1}^m a_{ik}^2\right)^{1/2}$
  Pour $i = 1, \ldots, m$
    $a_{ik} := a_{ik}/r_{kk}$
  Pour $j = k+1, \ldots, n$
    $r_{kj} := \sum_{i=1}^m a_{ik} a_{ij}$
    Pour $i = 1, \ldots, m$
      $a_{ij} := a_{ij} - a_{ik} r_{kj}$ $\square$

L'algorithme requiert mn^2 flops.

Remarque 7.10 GSM s'adapte au problème des moindres carrés; il est aussi stable que Householder, mais un peu plus coûteux que ce dernier puisqu'il renouvelle $m-k$ vecteurs à la $k^{\text{ième}}$ étape. D'autre part, GSM est deux fois plus efficace que Householder pour construire une base orthogonale de $R(A)$ mais au prix d'une stabilité inférieure. En effet, l'erreur rétrograde de GSM donne

$$\hat{Q}_1^T \hat{Q}_1 = I + E_{\text{gsm}}, \qquad \|E_{\text{gsm}}\|_2 \approx \mathbf{u}\kappa_2(A),$$

alors que Householder donne

$$\hat{Q}_1^T \hat{Q}_1 = I + E_{\text{hous}}, \qquad \|E_{\text{hous}}\|_2 \approx \mathbf{u}.$$

On conclut que l'on peut employer GSM avec confiance pour résoudre le problème de la base orthonormée uniquement si les colonnes de A sont suffisamment indépendantes.

Exemple 7.5 *Soit*

$$A = \begin{bmatrix} 1 & 1 \\ 10^{-3} & 0 \\ 0 & 10^{-3} \end{bmatrix}, \qquad \kappa_2(A) \approx 1.4 \times 10^3.$$

Avec $\beta = 10, t = 6$ *et l'arithmétique tronquée, GSM donne*

$$[\hat{q_1}, \hat{q_2}] = \begin{bmatrix} 1 & 0 \\ 10^{-3} & -0.707\,107 \\ 0 & 0.707\,107 \end{bmatrix}.$$

7.3 Orthogonalisation de Givens

7.4 Méthode QR avec pivotage sur les colonnes

Soit $A \in \mathbf{R}^{m\times n}$. Si $\operatorname{rg} A = r < n \leq m$, alors dans la décomposition $A = QR$, un des $r_{kk} = 0$ et la méthode flanche lors de la substitution arrière pour $Q^T A x_{\mathrm{mc}} = Q^T b$. D'autre part, le problème des moindres carrés admet une infinité de solutions si $\operatorname{rg}(A) < n$. On pose les questions suivantes.

(1) Comment trouver une solution au problème des moindres carrés?

(2) Comment trouver x_{mc}?

(3) Comment faire les calculs alors que $\operatorname{cond}(A) = +\infty$?

Le but de la décomposition QR est de former une base orthonormée $\{q_1, q_2, \ldots, q_r\}$ de l'image de A, $R(A)$. En effet, si

$$R(A) = R(Q_1), \qquad Q_1 = [q_1, q_2, \ldots, q_r], \qquad \operatorname{rg}(A) = r,$$

alors

$$A = Q_1 S, \qquad S \in \mathbf{R}^{r\times n}$$

et

$$\|Ax - b\|_2 = \|Q_1 S x - b\|_2.$$

Toute solution x du système sous-déterminé

$$Sx = Q_1^T b$$

résout le problème des moindres carrés. Mais si $\operatorname{rg}(A) < n$, la factorisation QR ne produit pas nécessairement une base orthonormée de $R(A)$.

Exemple. Soit la décomposition QR de A:

$$A = [a_1, a_2, a_3] = [q_1, q_2, q_3] \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix};$$

alors $\mathrm{rg}(A) = 2$ et

$$R(A)\,\mathrm{lin}\{q_i, q_j\}, \qquad i \neq j.$$

Heureusement, l'algorithme 7.2 de Householder avec pivotage sur les colonnes donne une base orthonormée de $R(A)$. L'algorithme modifié calcule

$$A\Pi = QR, \qquad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix} \tag{7.19}$$

$$\begin{matrix} r & n-r \end{matrix}$$

où $\mathrm{rg}(A) = \mathrm{rg}(R_{11}) = r < n \leq m$, $Q \in \mathbf{R}^{m\times m}$ est orthogonale, R_{11} est triangulaire supérieure et Π est une permutation.
Si

$$A\Pi = [a_{c_1}, \ldots, a_{c_n}], \qquad Q = [q_1, \ldots, q_m],$$

alors pour $k = 1, \ldots, n$, on a

$$a_{c_k} = \sum_{i=1}^{\min\{r,k\}} r_{ik} q_i \in \mathrm{lin}\{q_1, \ldots, q_r\};$$

donc

$$R(A) = \mathrm{lin}\{q_1, \ldots, q_r\}.$$

La décomposition (7.19) nous donne la solution du problème des moindres carrés:

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\| \left(Q^T A\Pi\right) \left(\Pi^T x\right) - Q^T b \right\|_2^2 \\ &= \|R_{11} y - (c - R_{12} z)\|_2^2 + \|d\|_2^2, \end{aligned}$$

où

$$\Pi^T x = \begin{bmatrix} y \\ z \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}$$

et

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix} .$$

Pour minimiser $\|Ax - b\|_2$ on doit prendre

$$x = \Pi \begin{bmatrix} R_{11}^{-1}(c - R_{12} z) \\ z \end{bmatrix}.$$

Si $z = 0$, on a la solution dite *fondamentale*

$$x_f = \Pi \begin{bmatrix} R_{11}^{-1} c \\ 0 \end{bmatrix}.$$

Du fait que x_f a au plus r composantes non nulles, Ax_f est une combinaison d'au plus r colonnes de A.

On voit que $x_f \neq x_{\mathrm{mc}}$ si $R_{12} \neq 0$; en effet,

$$\|x_{\mathrm{mc}}\|_2 = \min_{z \in \mathrm{R}^{n-r}} \left\| x_f - \begin{bmatrix} R_{11}^{-1} R_{12} \\ -I_{n-r} \end{bmatrix} z \right\|_2 .$$

On peut montrer, à partir de cette caractérisation de x_{mc}, que

$$1 \leq \frac{\|x_f\|_2}{\|x_{\mathrm{mc}}\|_2} \leq \sqrt{1 + \|R_{11}^{-1} R_{12}\|_2^2}.$$

Les matrices Q et Π de la décomposition (7.19),

$$Q = Q_1 Q_2 \cdots Q_r, \quad \Pi = \Pi_1 \Pi_2 \cdots \Pi_r,$$

s'obtiennent de la façon qui suit:

Supposons que pour k on ait déjà calculé

$$\begin{aligned} (Q_{k-1} \cdots Q_1) A (\Pi_1 \cdots \Pi_{k-1}) &= R^{(k-1)} \\ &= \begin{bmatrix} R_{11}^{(k-1)} & R_{12}^{(k-1)} \\ 0 & R_{22}^{(k-1)} \end{bmatrix} \begin{matrix} k-1 \\ m-k+1 \end{matrix} , \\ & \qquad\quad k-1 \quad n-k+1 \end{aligned}$$

où $R_{11}^{(k-1)}$ est triangulaire supérieure. Notons

$$R_{22}^{(k-1)} = \left[v_k^{(k-1)}, \ldots, v_n^{(k-1)} \right]$$

et définissons le plus petit indice $p, k \leq p \leq n$, par la stratégie

$$\left\| v_p^{(k-1)} \right\|_2 = \max \left\{ \left\| v_k^{(k-1)} \right\|_2, \ldots, \left\| v_n^{k-1)} \right\|_2 \right\}. \tag{7.20}$$

Si $k = \mathrm{rg}(A) + 1$, le max $= 0$ et on a fini. Sinon, soit Π_k la permutation des colonnes p et k. On choisit une réflexion de Householder Q_k telle que dans le produit

$$R^{(k)} = Q_k R^{(k-1)} \Pi_k$$

on ait

$$\begin{bmatrix} r_{k+1,k}^{(k)} \\ \vdots \\ r_{mk}^{(k)} \end{bmatrix} = 0,$$

c'est-à-dire Π_k place la plus grande colonne de $R_{22}^{(k-1)}$ en norme dans la 1$^{\text{ière}}$ colonne et Q_k annule ses éléments sous la diagonale.

Remarque 7.11 Les normes se calculent simplement: si Z est orthogonale,

$$Z^T v = \begin{bmatrix} \alpha \\ w \end{bmatrix} \begin{matrix} 1 \\ s-1 \end{matrix} \Rightarrow \|w\|_2^2 = \|v\|_2^2 - \alpha^2.$$

Donc on requiert $O(mn)$ flops au lieu de $O\left(mn^2\right)$.

Algorithme 7.5 (QR avec pivotage sur les colonnes) *Soit $A \in \mathbf{R}^{m\times n}$. L'algorithme calcule la décomposition $A\Pi = QR$ définie par (*). La permutation $\Pi = [e_{c_1}, \ldots, e_{c_n}]$ est déterminée par la stratégie (**). L'élément r_{ij} renouvelle a_{ij} $(i \le j)$.*

$c_j := j \qquad (j = 1, \ldots, n)$
$\gamma_j := \sum_{i=1}^{m} a_{ij}^2 \qquad (j = 1, \ldots, n)$
Pour $k = 1, \ldots, n$
  Choisir $p \quad (k \le p \le n)$ *tel que* $\gamma_p = \max_{k\le j\le n} \gamma_j$.
  Si $\gamma_j = 0$
    alors *sortir*
    sinon
      Echanger c_k *et* c_p, γ_k *et* γ_p *pour* $i = 1, \ldots, m$.
      Déterminer $\tilde{Q}_k$ *de Householder telle que*

$$\tilde{Q}_k \begin{bmatrix} a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{mk} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

      $A := \operatorname{diag}\left(I_{k-1}, \tilde{Q}_k\right) A$
      $\gamma_j := \gamma_j - a_{kj}^2 \qquad (j = k+1, \ldots, n)$ □

L'algorithme requiert $2mnr - r^2(m+n) + 2r^3/3$ flops, où $r = \operatorname{rg}(A)$. Comme pour l'algorithme 7.2 de Householder, il est possible d'emmagasiner la matrice orthogonale Q en facteurs dans la partie de A sous la diagonale .

Exemple 7.6 *Soient $\beta = 10$, $t = 3$ et*

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 5 & 6 \\ 1 & 8 & 9 \\ 1 & 11 & 12 \end{bmatrix}.$$

L'application de l'algorithme 7.5 donne

$$\Pi = [e_3, e_2, e_1], \quad \hat{R} = \begin{bmatrix} -16.400 & -14.600 & -1.820 \\ 0 & 0.816 & -0.816 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

et

$$\hat{Q} = \begin{bmatrix} -0.182 & -0.816 & 0.514 & 0.191 \\ -0.365 & 0.408 & -0.827 & 0.129 \\ 0.548 & 0 & 0.113 & -0.829 \\ -0.730 & 0.408 & 0.200 & 0.510 \end{bmatrix}.$$

On vient de voir que la méthode QR avec pivotage sur les colonnes donne la solution fondamentale x_f et non la solution de norme 2 minimum x_{mc}. Pour obtenir cette dernière, il faut annuler R_{12} dans

$$Q^T A\Pi = \begin{array}{c} \left[\begin{array}{cc} R_{11} & R_{12} \\ 0 & 0 \end{array} \right] \\ r \quad n-r \end{array} \begin{array}{c} r \\ m-r \end{array}, \qquad r = \mathrm{rg}(A).$$

Au moyen de réflexions de Householder, on obtient

$$Z_r \cdots Z_2 Z_1 \left[\begin{array}{c} R_{11}^T \\ R_{12}^T \end{array} \right] = \begin{array}{c} \left[\begin{array}{c} T^T \\ 0 \end{array} \right] \\ \end{array} \begin{array}{c} r \\ n-r \end{array},$$

où T est triangulaire supérieure. Ce calcul requiert $r^2(n-r)$ flops s'il est bien organisé.

Posons

$$Z = \Pi Z_1 \cdots Z_r;$$

alors

$$Q^T A Z = \begin{array}{c} \left[\begin{array}{cc} T & 0 \\ 0 & 0 \end{array} \right] \\ r \quad n-r \end{array} \begin{array}{c} r \\ m-r \end{array}.$$

On obtient enfin x_{mc}:

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\| Q^T A Z Z^T x - Q^T b \right\|_2^2 \\ &= \|Tw - c\|_2^2 + \|d\|_2^2, \end{aligned}$$

où

$$x = Z \begin{array}{c} \left[\begin{array}{c} w \\ y \end{array} \right] \\ \end{array} \begin{array}{c} r \\ n-r \end{array}, \quad Q^T b = \begin{array}{c} \left[\begin{array}{c} c \\ d \end{array} \right] \\ \end{array} \begin{array}{c} r \\ m-r \end{array}.$$

Donc, avec $w = T^{-1}c$,

$$x_{\mathrm{mc}} = Z \left[\begin{array}{c} T^{-1}c \\ 0 \end{array} \right].$$

Définition 7.3 *Soit $A \in \mathbf{R}^{m\times n}$ et soient deux matrices orthogonales $Q \in \mathbf{R}^{m\times m}$ et $Z \in \mathbf{R}^{n\times n}$ telles que*

$$\left(Q^T A Z\right)_{ij} = 0 \quad \forall i > \mathrm{rg}(A), j > \mathrm{rg}(A).$$

On dit alors que

$$B = Q^T A Z = \begin{array}{c} \left[\begin{array}{cc} T & 0 \\ 0 & 0 \end{array} \right] \\ r \quad n-r \end{array} \begin{array}{c} r \\ m-r \end{array}.$$

est une décomposition orthogonale complète *de A.*

On voit que, étant donnée une décomposition orthogonale complète quelconque de A, on obtient la solution de norme 2 minimum x_{mc} du problème des moindres carrés par la résolution d'un système linéaire carré et régulier.

7.5 Méthode de la décomposition en valeurs singulières

La décomposition selon les valeurs singulières est une décomposition orthogonale complète. Il faut recourir à cette dernière dans les cas où l'algorithme 7.5 de la décomposition QR ne détecte pas le rang exact de A.

7.6 Étalonnage et équilibrage des systèmes et affinage des solutions

Chapitre 8

Problème spectral des matrices nonsymétriques

8.1 Théorie de la perturbation

8.2 Méthode de la puissance

8.3 Décomposition de Hessenberg et de Schur

8.4 Algorithme QR

8.5 Algorithme QZ et le problème spectral généralisé

Chapitre 9

Problème spectral des matrices symétriques

9.1 Théorie des perturbations

9.2 Tridiagonalisation et algorithme QR symétrique

9.3 Méthode de Jacobi

9.4 Problème spectral généralisé

Chapitre 10

Méthodes de récurrence pour les systèmes linéaires

10.1 Méthodes de Jacobi et de Gauss–Seidel

10.2 Méthode du gradient conjugué

Chapitre 11

Matrices tridiagonales

Travail de Nathalie Lowe, présenté à Rémi Vaillancourt le 20 décembre 1989 dans le cours MAT 4781.

11.1 Introduction

Les matrices tridiagonales jouent un rôle important, autant en pratique qu'en théorie; il est donc nécessaire de leur apporter un intérêt particulier.

La forme compacte qu'offre la matrice tridiagonale amène une réduction importante du nombre d'opérations arithmétiques lors du calcul des valeurs propres et des vecteurs propres, ainsi que pour la résolution du système d'équations linéaires $Ax = b$.

De plus, les matrices symétriques, que l'on retrouve en abondance dans des applications en physique, mécanique, génie électrique etc., peuvent être réduites à une forme tridiagonale et la méthode la plus efficace pour trouver toutes les valeurs propres d'une matrice symétrique avec ou sans les vecteurs propres associés est une combinaison de la tridiagonalisation de Householder et de l'algorithme QR avec translation.

Avant de se pencher sur les problèmes de la tridiagonalisation d'une matrice symétrique, de trouver sa décomposition LU afin de résoudre le système $Ax = b$ et de calculer les valeurs propres, il faudrait définir ce que l'on entend par une matrice tridiagonale.

Définition 11.1 *Soit une matrice d'ordre n, $A \in \mathbf{R}^{n\times n}$. A est dite* tridiagonale *si les seuls éléments non nuls se trouvent sur la diagonale ou adjacents à la diagonale, c'est-à-dire $a_{ij} = 0$, $|i - j| > 1$.*

Schématiquement A est de la forme:

$$\begin{bmatrix} \times & \times & 0 & \cdots & 0 \\ \times & \times & \times & \ddots & 0 \\ 0 & \times & \times & \times & \vdots \\ \vdots & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & \times & \times \end{bmatrix}.$$

11.2 Tridiagonalisation d'une matrice symétrique

On veut transformer une matrice symétrique en une matrice tridiagonale sans modifier les valeurs propres et avoir des vecteurs propres equivalents entre les deux matrices.

Il existe plusieurs méthodes qui permettent une telle tridiagonalisation mais ici nous n'étudierons que celle de Householder. La méthode de Householder est une suite de transformations orthogonales. Donc la matrice symétrique $A_1 = A \in \mathbf{R}^{n\times n}$ est réduite à la matrice tridiagonale (symétrique) A_{n-1} par $n-2$ transformations orthogonales:

$$A_{i+1} = P_i^T A_i P_i, \qquad i = 1, \ldots, n-2,$$

où

$$P_i = I - \frac{v_i v_i^T}{h_i}, \qquad h_i = \frac{1}{2} v_i^T v_i$$

$$v_i^T = \left[0, \ldots, 0, a_{i+1,i}^{(i)} + \operatorname{sgn}(a_{i+1,i}^{(i)})\sigma_i, a_{i+2,i}^{(i)}, a_{i+3,i}^{(i)} \ldots, a_{n,i}^{(i)}\right]$$

et

$$\sigma_i^2 = \left(a_{i+1,i}^{(i)}\right)^2 + \left(a_{i+2,i}^{(i)}\right)^2 + \ldots + \left(a_{n,i}^{(i)}\right)^2.$$

Pour mieux voir ces transformations, on considère

$$A = A_1 = \begin{bmatrix} \alpha_1 & c_1^T \\ c_1 & M_1 \end{bmatrix}$$

où $\alpha_1 = a_{11}$; alors

$$A_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1^T \end{bmatrix} \begin{bmatrix} \alpha_1 & c_1^T \\ c_1 & M_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \tilde{P}_1 \end{bmatrix} = P_1^T A P_1$$

où l'on choisit $\tilde{P}_1$ tel que $\tilde{P}_1^T c_1 = \beta_1 e_1$, $\tilde{e}_1 \in \mathbf{R}^{n-1}$. Donc

$$A_2 = \begin{bmatrix} \alpha_1 & \beta_1 \tilde{e}_1^T \\ \beta_1 \tilde{e}_1 & \tilde{P}_1^T M \tilde{P}_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & c_2^T \\ 0 & c_2 & M_2 \end{bmatrix}$$

et

$$\begin{aligned} A_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{P}_2^T \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & c_2^T \\ 0 & c_2 & M_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \tilde{P}_2 \end{bmatrix} \\ &= P_2^T A_2 P_2 = P_2^T P_1^T A_1 P_1 P_2, \end{aligned}$$

où, en général, pour la $i^{\text{ème}}$ transformation

$$P_i = \begin{bmatrix} I_i & 0 \\ 0 & \tilde{P}_i \end{bmatrix}$$

Enfin, nous avons

$$\begin{aligned} T &= P_{n-2}^T P_{n-3}^T \cdots P_2^T P_1^T A P_1 P_2 \cdots P_{n-2} \\ &= P^T A P, \end{aligned}$$

où P est orthogonale puisqu'elle est le produit de matrices orthogonales.

Exemple 11.1 *Soient la matrice symétrique et le vecteur:*

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 2 \\ 4 & 2 & 3 \end{bmatrix}, \qquad c_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

Alors

$$\sigma_1^2 = 9 + 16 = 25, \qquad v_1 = \begin{bmatrix} 3+5 \\ 4 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$$

et

$$h_1 = \frac{1}{2} v_1^T v_1 = \frac{1}{2} \times 80 = 40.$$

Donc

$$\tilde{P}_1 = I_2 - \frac{1}{h_1}\left(v_1 v_1^T\right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{40} \begin{bmatrix} 64 & 32 \\ 32 & 16 \end{bmatrix} = \begin{bmatrix} -3/5 & -4/5 \\ -4/5 & 3/5 \end{bmatrix},$$

$$P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3/5 & -4/5 \\ 0 & -4/5 & 3/5 \end{bmatrix}$$

et

$$\begin{aligned} A_2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3/5 & -4/5 \\ 0 & -4/5 & 3/5 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 2 \\ 4 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -3/5 & -4/5 \\ 0 & -4/5 & 3/5 \end{bmatrix} \\ &= \begin{bmatrix} 2 & -5 & 0 \\ -5 & 132/25 & 26/25 \\ 0 & 26/25 & 43/25 \end{bmatrix} = T, \end{aligned}$$

où T est tridiagonale. □

On peut modifier cette méthode en utilisant le fait bien important que $\tilde{P}_k M_k \tilde{P}_k$ est symétrique pour chaque k. Puisque $\tilde{P}_k$ est de la forme

$$\tilde{P}_k = I - \frac{2}{v^T v} v v^T, \qquad v \neq 0 \in \mathbf{R}^{n-k},$$

posons

$$\tilde{P} = \frac{2}{v^T v} M_k v \;\text{ et }\; w = \tilde{P} - \frac{p^T v}{v^T v} v;$$

alors on peut écrire

$$\tilde{P}_k M_k \tilde{P}_k = M_k - v w^T - w v^T$$

où $v = v_k$ et, à la $k^{\text{ième}}$ étape, nous avons la matrice

$$\begin{aligned} A_k = P_{k-1} \cdots P_1 A P_1 \cdots P_{k-1} &= \begin{bmatrix} & B & & 0 \\ & & & c_{k-1}^T \tilde{P}_{k-1} \\ 0 & \tilde{P}_{k-1} c_{k-1} & & \tilde{P}_{k-1} M_{k-1} \tilde{P}_{k-1} \end{bmatrix} \\ &= \begin{bmatrix} & B & & 0 \\ & & & \beta_{k-1} e_1^T \\ 0 & \beta_{k-1} e_1 & & M_{k-1} - v w^T - w v^T \end{bmatrix}, \end{aligned}$$

où B est tridiagonale.

Remarque 11.1 La matrice tridiagonale T a les propriétés suivantes:

(a) T est symétrique:

$$T^T = \left(P^T A P\right)^T = P^T A^T P = P^T A P = T;$$

(b) T admet les mêmes valeurs propres que A:

$$\begin{aligned} \det(T - \lambda I) &= \det(P^T A P - \lambda I) = \det(P^T (A - \lambda I) P) \\ &= \det P^T \det(A - \lambda I) \det P = \det(A - \lambda I); \end{aligned}$$

(c) Si z est un vecteur propre de T, alors Pz est un vecteur propre de A:

$$Tz = \lambda z, \qquad (P^T A P) z = \lambda z, \qquad A(Pz) = \lambda (Pz);$$

(d) Si y^* est une solution du système $Ty = P^T b$, alors Py^* est une solution de $Ax = b$:

$$Ty^* = P^T b, \qquad P^T A P y^* = P^T b, \qquad A(Py^*) = b.$$

Il existe donc une relation entre les vecteurs propres d'un matrice symétrique A et ceux de la matrice tridiagonale T obtenue par la méthode de Householder; d'autre part les valeurs propres restent inchangées. Il existe aussi une relation entre la solution du système $Ax = b$ et celle du système $Ty = P^T b$.

Si un problème contient une matrice symétrique, nous n'avons qu'à travailler avec la matrice tridiagonale associée, et ensuite faire les transformations nécessaires.

L'algorithme de la tridiagonalisation de Householder, tel que présenté par J. H. Wilkinson, se retrouve à l'annexe A.

11.3 Décomposition LU

Pour illustrer la simplicité de la décomposition LU d'une matrice tridiagonale, considérons l'exemple:

$$A = \begin{bmatrix} 2 & 4 & 0 & 0 \\ 4 & 5 & -6 & 0 \\ 0 & -6 & -4 & -4 \\ 0 & 0 & -4 & 2 \end{bmatrix}.$$

La décomposition LU produit:

$$A = LU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 0 & 0 \\ 0 & -3 & -6 & 0 \\ 0 & 0 & 8 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Remarquons que la co-diagonale supérieure de U est celle de A. On peut voir que chaque étape du calcul de U s'effectue sur une sous-matrice 2×2. Vérifions ce fait.

$$\begin{aligned} A &= \begin{bmatrix} 2 & 4 & 0 & 0 \\ 4 & 5 & -6 & 0 \\ 0 & -6 & 4 & -2 \\ 0 & 0 & -4 & 2 \end{bmatrix} \quad 1^{\text{ière}} \text{ étape } \begin{bmatrix} 2 & 4 \\ 4 & 5 \end{bmatrix} \sim \begin{bmatrix} 2 & 4 \\ 0 & -3 \end{bmatrix} \\ &\sim \begin{bmatrix} 2 & 4 & 0 & 0 \\ 0 & -3 & -6 & 0 \\ 0 & -6 & -4 & -4 \\ 0 & 0 & -4 & 2 \end{bmatrix} \quad 2^{\text{ème}} \text{ étape } \begin{bmatrix} -3 & -6 \\ -6 & -4 \end{bmatrix} \sim \begin{bmatrix} -3 & -6 \\ 0 & 8 \end{bmatrix} \\ &\sim \begin{bmatrix} 2 & 4 & 0 & 0 \\ 0 & -3 & -6 & 0 \\ 0 & 0 & 8 & -4 \\ 0 & 0 & -4 & 2 \end{bmatrix} \quad 3^{\text{ème}} \text{ étape } \begin{bmatrix} 8 & -4 \\ -4 & 2 \end{bmatrix} \sim \begin{bmatrix} 8 & -4 \\ 0 & 0 \end{bmatrix} \\ &\sim \begin{bmatrix} 2 & -4 & 0 & 0 \\ 0 & -3 & -6 & 0 \\ 0 & 0 & 8 & -4 \\ 0 & 0 & 0 & 0 \end{bmatrix} = U. \end{aligned}$$

Cette méthode, qui fonctionne même pour les matrices non symétriques, est donc très simple.

Algorithme 11.1 *Soit la matrice tridiagonale*

$$A = \begin{bmatrix} d_1 & u_1 & 0 & \cdots & 0 \\ l_1 & d_2 & u_2 & & 0 \\ 0 & l_2 & d_3 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & u_{n-1} \\ 0 & \cdots & 0 & l_{n-1} & d_n \end{bmatrix}.$$

L'algorithme renouvelle A par la décomposition Lu de A:

do $k = 1$ *à* $n-1$
  $l_k \leftarrow l_k/d_k$
  $d_{k+1} \leftarrow d_{k+1} - l_k u_k$
end do

L'algorithme prend environ $3(n-1)$ flops. On voit que

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_1 & 1 & 0 & & 0 \\ 0 & l_2 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} d_1 & u_1 & 0 & \cdots & 0 \\ 0 & d_2 & u_2 & & \vdots \\ 0 & 0 & \ddots & \ddots & 0 \\ \vdots & & \ddots & d_{n-1} & u_{n-1} \\ 0 & \cdots & 0 & 0 & d_n \end{bmatrix}.$$

Pour résoudre le système $Ax = b \Rightarrow LUx = b$ on résout $Ly = b$ par substitutions avant et ensuite $Ux = y$ par substitutions arrière.
Pour résoudre $Ly = b$, on emploie la récurrence

$$y_k = b_k - l_{k-1} y_{k-1}, \qquad y_1 = b_1.$$

On a donc l'algorithme suivant.

Algorithme 11.2
do $k = 2$ *à* n
  $b_k \leftarrow b_k - l_{k-1} b_{k-1}$
end do

Cet algorithme prend environ $2(n-2)$ flops.

Pour résoudre $Ux = y$, on emploie la récurrence

$$x_k = (y_k - u_k x_{k+1})/d_k, \qquad x_n = y_n/d_n,$$

qui produit l'algorithme

Algorithme 11.3
$y_n \leftarrow y_n/d_n$
do $k = n-1$ *à* 1
  $y_k \leftarrow (y_k - u_k y_{k+1})/d_k$
end do

Cet algorithme prend environ $3(n-1)+1$ flops.

En combinant les trois algorithmes précédents, on peut résoudre le système $Ax = b$, où A est une matrice tridiagonale, le tout prenant $8n-6$ flops. Ceci représente une réduction importante du nombre d'opérations puisque pour résoudre $Ly = b$ avec une matrice pleine, on a besoin de $O\left(n^2\right)$ flops.

Remarque 11.2 Pour la décomposition QR la méthode est semblable mais naturellement avec moins d'opérations à affectuer ($O(n)$ flops). Il est intéressant de noter que le résultat final produit une matrice Q Hessenberg supérieure.

11.4 Valeurs propres

Nous allons traiter, ici, de deux méthodes pour obtenir les valeurs propres d'une matrice tridiagonale symétrique. La première est la méthode QR avec la translation optimum de Wilkinson. La deuxième est la méthode de la bisection, particulière aux matrices tridiagonales; elle utilise les suites de Sturm. Il faut d'abord définir une matrice non réduite.

Définition 11.2 *La matrice tridiagonale symétrique*

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \vdots \\ 0 & \beta_2 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_n \end{bmatrix} \tag{11.1}$$

est dite non réduite *si* $\beta_i \neq 0$, $i = 1, \ldots, n-1$.

Il suffit de traiter le cas non réduit; en effet si $\beta_i = 0$, le problème se découple en deux matrices tridiagonales de la façon suivante:

$$\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}.$$

L'ensemble des valeurs propres de T, c'est-à-dire le spectre de T, noté $\lambda(T)$, et les spectres de T_1 et T_2 satisfont la relation:

$$\lambda(T) = \lambda(T_1) \cup \lambda(T_2).$$

Nous supposons donc que toutes les matrices sont tridiagonales symétriques et non réduites. De plus, pour simplifier les énoncés des résultats, nous supposons que les valeurs propres de S sont bien étalées:

$$|\lambda_i| > |\lambda_{i+1}|, \qquad i = 1, \ldots, n-1. \tag{11.2}$$

11.4.1 La méthode QR avec translation.

Rappellons que la méthode QR sans translation

$$\begin{aligned} A = A_0 &= Q_1 R_1 \\ A_1 &= R_1 Q_1 \\ &= Q_2 R_2 \\ &\vdots \\ A_\infty &= QR \end{aligned}$$

converge vers une matrice triangulaire supérieure A_∞ et $\lambda(A_\infty) = \lambda(A)$. Sous l'hypothèse (11.2), la convergence est d'ordre 2.

La méthode QR avec la translation μ:

$$\begin{aligned} T &= A_0 \\ A_0 - \mu I &= Q_1 R_1 \\ A_1 &= R_1 Q_1 + \mu I \\ &\vdots \\ H &= RQ + \mu I \end{aligned}$$

converge vers une matrice triangulaire supérieure et $\lambda(H) = \lambda(T)$. On a l'algorithme suivant.

Algorithme 11.4
$T_0 = Q_0^T A Q_0$ *(tridiagonal)*
$k = 1, 2, \ldots,$
$T_{k-1} - \mu I = Q_k R_k$ *(décomposition QR)*
$T_k := R_k Q_k + \mu I$

La tridiagonalité est préservée, puisque la structure Hessenberg et la symétrie sont préservées.

On choisit la translation μ pour accélérer la convergence. On peut prendre $\mu = \alpha_n$. Mais pour les matrices tridiagonales il existe un meilleur choix basé sur la valeur propre de la matrice:

$$\begin{bmatrix} \alpha_{n-1} & \beta_{n-1} \\ \beta_{n-1} & \alpha_n \end{bmatrix}.$$

On appelle cette translation, la translation de Wilkinson; elle est donnée par

$$\mu = \alpha_n + d - \operatorname{sgn}(d)\sqrt{d^2 + \beta_{n-1}^2},$$

où $d = (\alpha_{n-1} - \alpha_n)/2$. Cette translation est optimum: elle donne à la méthode QR une convergence d'ordre trois.

La mise en œuvre de Pal–Walker–Kahan de l'algorithme QR avec la translation de Wilkinson implicite utilise le fait que cet algorithme préserve les similarités. Donc, on peut toujours écrire

$$T_k = Q_k^T T_{k-1} Q_k.$$

En utilisant la rotation de Givens:

$$G_1 = G(1, 2, \theta) \Rightarrow \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} \alpha_1 - \mu \\ \beta_1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix},$$

nous avons $G_1 e_1 = Q_1 e_1$, c'est-à-dire, G_1 et Q_1 ont la même $1^{\text{ière}}$ colonne. Mais

G_1 introduit des éléments non nuls en positions $(3,1)$ et $(1,3)$:

$$T_1 = G_1^T T_0 G_1 = \begin{bmatrix} \times & \times & + & 0 & \cdots & \\ \times & \times & \times & 0 & \cdots & \\ + & \times & \times & \times & 0 & \\ 0 & 0 & \times & \times & \times & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

On chasse les deux + vers le bas au moyen de $G_2 = G(2,3,\theta_2)$:

$$T_2 = G_2^T T_1 G_2 = \begin{bmatrix} \times & \times & 0 & 0 & \cdots & \\ \times & \times & \times & + & \cdots & \\ 0 & \times & \times & \times & 0 & \\ 0 & + & \times & \times & \times & \ddots \\ \vdots & 0 & 0 & \ddots & \ddots & \ddots \end{bmatrix}$$

etc. A la $i^{\text{ième}}$ étape, T_i n'est pas tridiagonale; il faut donc compléter les $n-1$ étapes (rotations) pour obtenir une première approximation. Ce processus est souvent appelé "chasing the bulge". A la fin, nous obtenons

$$\begin{aligned} T &= T_0 \\ T_1 &= G_1^T T_0 G_1 \\ T_2 &= G_2^T (G_1^T T_0 G_1) G_2 \\ &\vdots \\ T_{n-1} &= G_{n-1}^T G_{n-2}^T \ldots G_2^T G_1^T T_0 G_1 G_2 \ldots G_{n-1} \\ &= Z^T T Z, \end{aligned}$$

où $Z^T T Z$ est tridiagonale. Puisque $Ze_1 = G_1 e_1 = Q_1 e_1$, on voit, par le théorème sur l'"unicité" de la décomposition de Hessenberg, que la matrice tridiagonale $Z^T T Z$ est essentiellement la matrice T qu'on obtient par la méthode QR avec translation.

Exemple 11.2 *Soit la matrice tridiagonale et symétrique*

$$T = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 0.01 \\ 0 & 0 & 0.01 & 4 \end{bmatrix}.$$

Alors

$$Z^T T Z = G_3^T G_2^T G_1^T T G_1 G_2 G_3 = \begin{bmatrix} 0.5 & 0.591\,6 & 0 & 0 \\ 0.591\,6 & 1.785 & 0.180\,8 & 0 \\ 0 & 0.180\,8 & 3.714 & 0.000\,004\,4 \\ 0 & 0 & 0.000\,004\,4 & 4.002\,497 \end{bmatrix};$$

mais les valeurs propres exactes sont

$$\{0.265\ldots, 2, 3.731\ldots, 4\}. \quad \square$$

Cet exemple montre que cette méthode est instable: en effet, l'erreur augmente lorsqu'on remonte le long de la diagonale; l'erreur dépend donc de l'ordre de la matrice.

11.4.2 La suite de Sturm d'une matrice tridiagonale.

On déconseille souvent de calculer le polynôme caractéristique pour trouver les valeurs propres, mais dans le cas d'une matrice tridiagonale symétrique, ce polynôme peut être calculé directement. La méthode de la bisection utilise des formules de récurrence qui calculent le polynôme caractéristique pour ensuite approximer les valeurs propres à l'aide de la propriété des suites de Sturm.

Soit $p_i(\lambda)$ le $i^{\text{ième}}$ mineur principal de la matrice $T - \lambda I$ et posons $p_0(\lambda) \equiv 1$; alors

$$\begin{aligned} p_1(\lambda) &= \alpha_1 - \lambda, \\ p_2(\lambda) &= (\alpha_2 - \lambda)p_1(\lambda) - \beta_1^2 = (\alpha_2 - \lambda)p_1(\lambda) - \beta_1^2 p_0(\lambda), \\ p_3(\lambda) &= (\alpha_3 - \lambda)p_2(\lambda) - \beta_2^2 p_1(\lambda). \end{aligned}$$

En général, nous avons le lemme suivant.

Lemme 11.1 *Soit $p_i(\lambda)$ le $i^{\text{ième}}$ mineur principal de la matrice triangulaire symétrique $T - \lambda I$. Si l'on pose $\beta_0 \equiv 0$ et $p_{-1}(\lambda) \equiv 0$, alors*

$$p_i(\lambda) = (\alpha_i - \lambda)p_{i-1}(\lambda) - \beta_{i-1}^2 p_{i-2}(\lambda), \qquad i = 1, 2, \ldots, n. \tag{11.3}$$

Démonstration. Il suffit de développer le déterminant:

$$p_i(\lambda) = \det \begin{bmatrix} \alpha_1 - \lambda & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 - \lambda & \beta_2 & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \alpha_{i-1} - \lambda & \beta_{i-1} \\ 0 & \cdots & 0 & \beta_{i-1} & \alpha_i - \lambda \end{bmatrix} \tag{11.4}$$

suivant la dernière ligne. $\square$

On remarque qu'on peut obtenir le polynôme caractéristique de T, $p_n(\lambda) = \det(T - \lambda)$, par ces formules de récurrence.

Nous allons maintenant définir une suite de Sturm.

Définition 11.3 *Une* suite de Sturm *est une suite de fonctions $\{f_n(x)\}$ telles que si $f_n(x_0) = 0$ alors $f_{n-1}(x_0)f_{n+1}(x_0) < 0$.*

On démontre le lemme suivant:

Lemme 11.2 *Les polynômes $p_i(\lambda)$ forment une suite de Sturm.*

Démonstration. Soit λ_0 une racine de $p_i(\lambda) = 0$, $1 \leq i \leq n-1$. Par l'équation générale

$$\begin{aligned} p_{i+1}(\lambda_0) &= (\alpha_{i+1} - \lambda_0)p_i(\lambda_0) - \beta_i^2 p_{i-1}(\lambda_0) \\ &= -\beta_i^2 p_{i-1}(\lambda_0). \end{aligned}$$

Puisque T est non réduite, $\beta_i \neq 0$; alors ou bien $p_{i+1}(\lambda_0)p_{i-1}(\lambda_0) < 0$ ou bien $p_{i-1}(\lambda_0) = 0$. Si $p_{i-1}(\lambda_0) = p_{i+1}(\lambda_0) = 0$, par la formule générale:

$$p_i(\lambda_0) = p_{i-1}(\lambda_0) = \ldots = p_1(\lambda_0) = p_0(\lambda_0) = 0.$$

Mais ceci est une absurdité car $p_0(\lambda) \equiv 1$. Donc $p_{i+1}(\lambda_0)p_{i-1}(\lambda_0) < 0$. □

Lemme 11.3 *Le polynôme p_i admet i zéros distincts et ceux-ci séparent les zéros du polynôme p_{i+1}.*

Démonstration. (Par induction) Puisque $p_1(\lambda)$ admet un zéro à α_1, alors $p_2(\alpha_1) = -\beta_1^2 < 0$ et $p_2(\lambda) = \lambda^2 + O(\lambda) \to +\infty$ si $|\lambda| \to \infty$ et donc p_2 a deux zéros, l'un $< \alpha_1$ et l'autre $> \alpha_1$.

Par induction, supposons que les zéros de $p_{i-2}(\lambda)$ et $p_{i-1}(\lambda)$ sont distincts et que ceux de p_{i-2} séparent ceux de p_{i-1} (v. fig. 11.1).

Soient $\lambda_1^{(i-1)} > \lambda_2^{(i-1)} > \cdots \lambda_{i-1}^{(i-1)}$ les zéros de p_{i-1} et donc, pour $k \leq i-1$,

$$p_i\left(\lambda_k^{(i-1)}\right) = -\beta_{i-1}^2 p_{i-2}\left(\lambda_k^{(i-1)}\right) \Rightarrow p_i\left(\lambda_k^{(i-1)}\right) p_{i-2}\left(\lambda_k^{(i-1)}\right) < 0.$$

Par hypothèse, p_{i-2} change de signe entre $\lambda_k^{(i-1)}$ et $\lambda_{k-1}^{(i-1)}$ pour $k = 2, \ldots, i-1$. Donc p_i change de signe entre $\lambda_k^{(i-1)}$ et $\lambda_{k-1}^{(i-1)}$; alors p_i a une racine entre chaque paire de racines adjacentes de p_{i-1}. De plus

$$p_i(\lambda) \to \begin{cases} \infty & \text{si } \lambda \to -\infty, \\ (-1)^i \times \infty & \text{si } \lambda \to +\infty. \end{cases} \tag{11.5}$$

Puisque $p_{i-2}\left(\lambda_{i-1}^{(i-1)}\right) > 0$, p_i admet une racine à gauche de $\lambda_{i-1}^{(i-1)}$. De même, puisque $p_{i-2}\left(\lambda_1^{(i-1)}\right) > 0$ si i est pair et < 0 si i est impair, alors p_i admet une racine à droite de $\lambda_1^{(i-1)}$. Donc p_i admet i racines distinctes. □

Par le lemme 11.3 et le fait que $p_n(\lambda)$ est le polynôme caractéristique d'une matrice tridiagonale symétrique T, dont les racines sont les valeurs propres de T, nous avons le lemme suivant.

Lemme 11.4 *Les valeurs propres d'une matrice tridiagonale symétrique non réduite sont distinctes.*

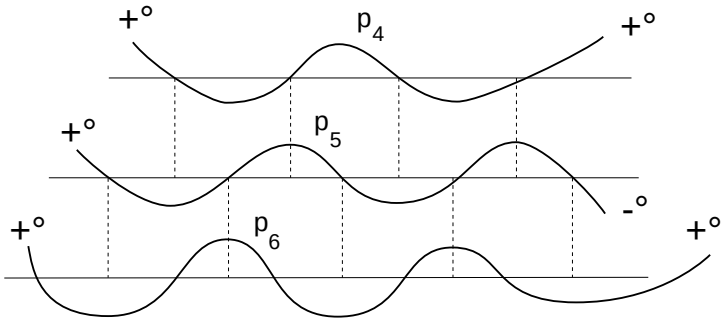

Figure 11.1: Racines de p_4, p_5 et p_6.

Nous pouvons maintenant passer au calcul des valeurs propres. Posons

$$\operatorname{sgn} p_i(\lambda) = \begin{cases} \operatorname{sgn} p_i(\lambda) & \text{si } p_i(\lambda) \neq 0, \\ \operatorname{sgn} p_{i-1}(\lambda) & \text{si } p_i(\lambda) = 0. \end{cases} \tag{11.6}$$

(Note: si $p_i(\lambda) = 0$, $p_{i-1}(\lambda) \neq 0$ car nous avons une suite de Sturm.)

Notation 11.1 *Soit l'ensemble*

$$E_i(\lambda) = \{+, \operatorname{sgn} p_1(\lambda), \operatorname{sgn} p_2(\lambda), \ldots, \operatorname{sgn} p_i(\lambda)\} \tag{11.7}$$

et notons $\sigma_i(\lambda)$ le nombre de paires consécutives de même signe de $E_i(\lambda)$.

Le premier signe positif vient du fait que $p_0(\lambda) \equiv 1$. Par exemple, si

$$E_{12}(\lambda) = \{+, -, -, +, -, +, +, +, -, +, +, -, -\},$$

alors $\sigma_{12}(\lambda) = 5$.

Théorème 11.1 *Pour $\lambda \in \mathbf{R}$ fixé, $\sigma_n(\lambda)$ est égal au nombre de racines de $p_n(\lambda) = \det(T - \lambda I)$ supérieures ou égales à λ.*

Démonstration. (Par induction) Pour $i = 1$

$$\begin{aligned} \lambda < \alpha_1 \quad &\Rightarrow \quad E_1(\lambda) = \{+, +\} \Rightarrow \sigma_1(\lambda) = 1, \\ \lambda = \alpha_1 \quad &\Rightarrow \quad E_1(\lambda) = \{+, +\} \Rightarrow \sigma_1(\lambda) = 1, \\ \lambda > \alpha_1 \quad &\Rightarrow \quad E_1(\lambda) = \{+, -\} \Rightarrow \sigma_1(\lambda) = 0. \end{aligned}$$

Supposons que l'énoncé est vrai pour $k = 1, 2, \ldots, i$; il faut le démontrer pour $k = i + 1$. Notons les racines de p_i:

$$\lambda_i^{(i)} < \lambda_{i-1}^{(i)} < \cdots < \lambda_1^{(i)}.$$

Il suit de la définition de $\sigma_i(\lambda)$ (notée σ_i ci-après pour simplifier), de l'hypothèse de récurrence

$$\lambda_i^{(i)} < \cdots < \lambda_{\sigma_{i+1}}^{(i)} < \lambda \leq \lambda_{\sigma_i}^{(i)} < \lambda_{\sigma_i - 1}^{(i)} \cdots < \lambda_1^{(i)},$$

et du lemme 11.3, que

$$\lambda^{(i)}_{\sigma_i+1} < \lambda^{(i+1)}_{\sigma_i+1} < \lambda^{(i)}_{\sigma_i}.$$

On a donc 4 cas possibles pour la position de λ:

(a) $\lambda = \lambda^{(i)}_{\sigma_i}$,

(b) $\lambda^{(i+1)}_{\sigma_i+1} < \lambda < \lambda^{(i)}_{\sigma_i}$,

(c) $\lambda = \lambda^{(i+1)}_{\sigma_i+1}$,

(d) $\lambda^{(i)}_{\sigma_i+1} < \lambda < \lambda\sigma_i + 1^{(i+1)}$.

Dans le cas (a), $\lambda = \lambda^{(i)}_{\sigma_i} \Rightarrow$

$$\begin{aligned} E_i(\lambda) &= \{+, \ldots, \operatorname{sgn} p_{i-1}(\lambda), \operatorname{sgn} p_{i-1}(\lambda)\}, \qquad \text{car } p_i(\lambda) = 0, \\ E_{i+1}(\lambda) &= \{+, \ldots, \operatorname{sgn} p_{i-1}(\lambda), \operatorname{sgn} p_{i-1}(\lambda), \operatorname{sgn} p_{i+1}(\lambda)\}, \end{aligned}$$

λ étant racine de p_i, $p_{i+1}(\lambda)$ et $p_{i-1}(\lambda)$ sont de signes contraires, et donc $\sigma_{i+1}(\lambda) = \sigma_i(\lambda) =$ nombre de racines de p_i qui sont $\geq \lambda$. Puisque p_i et p_{i+1} ont le même nombre de racines $\geq \lambda$,

$$\sigma_{i+1}(\lambda) \text{ est bien le nombre de racines de } p_{i+1} \geq \lambda.$$

Dans le cas (b), $\lambda^{(i+1)}_{\sigma_i+1} < \lambda < \lambda^{(i)}_{\sigma_i} \Rightarrow$

$$\begin{aligned} E_i(\lambda) &= \{+, \ldots, \operatorname{sgn} p_i(\lambda)\}, \\ E_{i+1}(\lambda) &= \{+, \ldots, \operatorname{sgn} p_i(\lambda), \operatorname{sgn} p_{i+1}(\lambda)\}. \end{aligned}$$

Par récurrence ou par la figure 11.1, on voit, qu'en une racine $\lambda^{(i)}_j$ de p_i,

$$\operatorname{sgn} p'_i(\lambda^{(i)}_j) = \operatorname{sgn} p_{i+1}(\lambda^{(i)}_j);$$

donc (v. fig. 11.2)

$$\operatorname{sgn} p_{i+1}(\lambda) = -\operatorname{sgn} p_i(\lambda) \Rightarrow \sigma_{i+1}(\lambda) = \sigma_i(\lambda).$$

Comme les polynômes p_{i+1} et p_i ont le même nombre de racines $\geq \lambda$, $p_{i+1}(\lambda)$ admet exactement $\sigma_{i+1}(\lambda)$ racines $\geq \lambda$.

Dans le cas (c), $\lambda = \lambda^{(i+1)}_{\sigma_{i+1}} \Rightarrow$

$$\begin{aligned} E_i(\lambda) &= \{+, \ldots, \operatorname{sgn} p_i(\lambda)\}, \\ E_{i+1}(\lambda) &= \{+, \ldots, \operatorname{sgn} p_i(\lambda), \operatorname{sgn} p_i(\lambda)\}, \end{aligned}$$

car $p_{i+1}(\lambda) = 0$. Donc $\sigma_{i+1}(\lambda) = \sigma_i(\lambda) + 1$ et comme le polynôme p_{i+1} a une racine de plus que p_i qui soit $\geq \lambda$, $\sigma_{i+1}(\lambda)$ est encore égal au nombre de racines de p_{i+1} qui sont $\geq \lambda$.

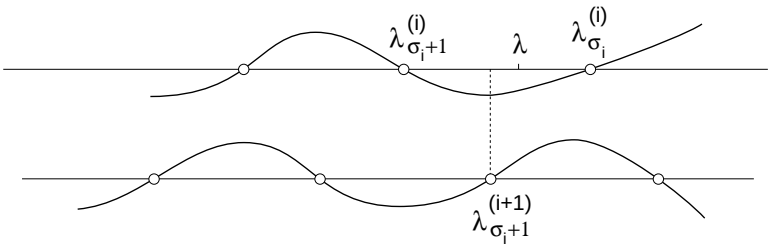

Figure 11.2: Signes de $p_{i+1}(\lambda)$ et de $p_i(\lambda)$.

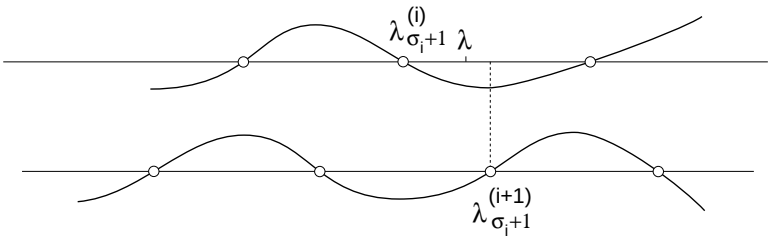

Figure 11.3: Le polynôme $p_{i+1}(\lambda)$ a une racine de plus que le polynôme $p_i(\lambda)$.

Dans le cas (d), $\lambda^{(i)}_{\sigma_i+1} < \lambda < \lambda^{(i)}_{\sigma i+1} \Rightarrow$

$$\begin{aligned} E_i(\lambda) &= \{+, \cdots, \operatorname{sgn} p_i(\lambda)\} \\ E_{i+1}(\lambda) &= \{+, \cdots, \operatorname{sgn} p_i(\lambda), \operatorname{sgn} p_{i+1}(\lambda)\} \end{aligned}$$

et comme dans le cas (c),

$$\operatorname{sgn} p_{i+1}(\lambda) = \operatorname{sgn} p_i(\lambda) \Rightarrow \sigma_{i+1}(\lambda) = \sigma_i(\lambda) + 1$$

(v. fig. 11.3). Et puisque le polynôme p_{i+1} admet une racine de plus que p_i à droite de λ, le résultat est valide. □

11.4.3 Le calcul des valeurs propres par la bisection.

Nous avons établi tous les outils nécessaires pour l'emploi de la méthode de la bisection. Rappelons que cette méthode détermine la racine d'une fonction continue en réduisant successivement les intervalles qui contiennent la racine. En voici la procédure.

On choisit un intervalle $[a_0, b_0]$ où $f(a)$ et $f(b)$ sont de signes opposés, disons, $f(a_0) < 0$ et $f(b_0) > 0$. Par le théorème des valeurs intermédiaires il existe un nombre $d \in [a_0, b_0]$ tel que $f(d) = 0$. On pose $c_0 = (a_0 + b_0)/2$. Si $f(c_0) > 0$ alors $d \in [a_0, c_0] =: [a_1, b_1]$ et si $f(c_0) < 0$ alors $d \in [c_0, b_0] =: [a_1, b_1]$. On continue ainsi à réduire de moitié les intervalles $[a_i, b_i]$, $i = 0, 1, \ldots$, qui tous encadrent la racine recherchée. La procédure est illustrée dans la figure 11.4.

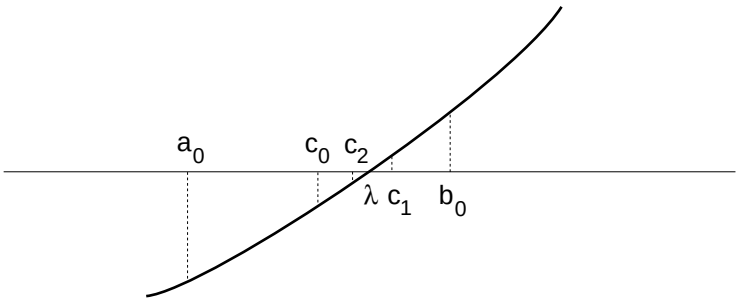

Figure 11.4: Deux pas de la méthode de la bisection.

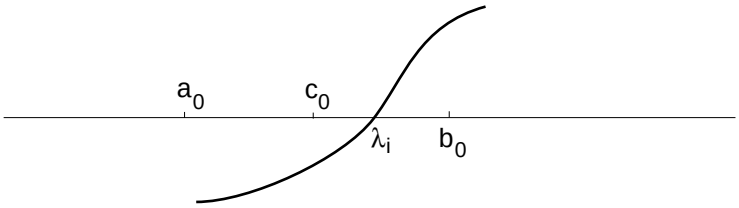

Figure 11.5: $\sigma_n(c_0) \geq i$.

Nous allons utiliser cette méthode pour déterminer les valeurs propres de T, qui sont en fait les racines du polynôme caractéristique.

Pour calculer la $i^{\text{ème}}$ valeur propre λ_i ($\lambda_1 > \cdots > \lambda_i > \cdots > \lambda_n$), soit $[a_0, b_0]$ un intervalle tel que

$$\lambda_i \in [a_0, b_0].$$

On peut prendre, par exemple, $-a_0 = b_0 = \|T\|_\infty$. On remarque que $\|T\|_1 = \|T\|_\infty$ parce que T est symétrique.

Soit $c_0 = (a_0 + b_0)/2$ le milieu de l'intervalle. On calcule $\sigma_n(c_0)$. Alors on a un des deux cas suivants:

$$\sigma_n(c_0) \geq i \quad \Rightarrow \quad \lambda_i \in [c_0, b_0]$$

(v. fig. 11.5);

$$\sigma_n(c_0) < i \quad \Rightarrow \quad \lambda_i \in [a_0, c_0]$$

(v. fig. 11.6).

On a donc $\lambda_i \in [a_1, b_1]$ où, dans le premier cas,

$$a_1 = c_0, \qquad b_1 = b_0,$$

et dans le deuxième cas,

$$a_1 = a_0, \qquad b_1 = c_0.$$

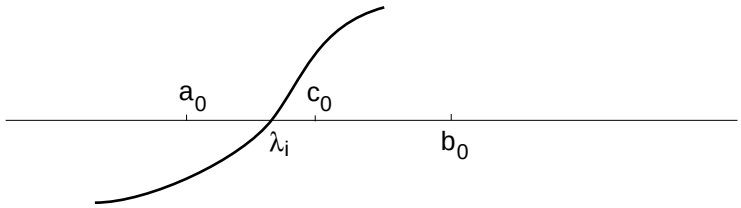

Figure 11.6: $\sigma_n(c_0) < i$.

On continue le procédé en réduisant successivement l'intervalle

$$[a_k, b_k], \qquad k = 0, 1, 2, \ldots,$$

tel que

$$\lambda_i \in [a_k, b_k]$$

et

$$b_k - a_k = \frac{b_0 - a_0}{2^k} \to 0, \text{ si } k \to \infty.$$

Pour appliquer la méthode de la bisection, il faut calculer $p_1(\lambda), p_2(\lambda), \ldots, p_n(\lambda)$ pour obtenir $\sigma_n(\lambda)$ avec $\lambda = c_0, c_1, \ldots, c_k, \ldots$. Si l'on choisit $b_0 = \|T\|_\infty$, alors

$$b_0 = \max_{1 \le i \le n} \{|\alpha_1| + |\beta_1|, \ldots, |\beta_{i-1}| + |\alpha_i| + |\beta_i|, \ldots, |\beta_{n-1}| + |\alpha_n|\}$$

et

$$a_0 = -b_0.$$

Mais le théorème de Gerschgorine donne des bornes plus précises en utilisant

$$r_1 = |\beta_1|, \qquad r_i = |\beta_{i-1}| + |\beta_i| \qquad r_n = |\beta_{n-1}|$$

et en définissant $a_0 = \min_i(\alpha_i - r_i)$, $b_0 = \max_i(\alpha_i + r_i)$.

La méthode de la bisection est stable. L'erreur sur la $i^{\text{ème}}$ valeur propre calculée en k étapes de bisection est de l'ordre de 2^{-k}, et donc elle est indépendante de n, le rang de T.

Pour calculer les n valeurs propres en effectuant k bisections, nous avons besoin de $2(2n)nk = 4kn^2$ flops puisque les formules de récurrence $p_1(\lambda), \ldots, p_n(\lambda)$ qui produisent $\sigma_n(\lambda)$, consistent en $2n$ multiplications et $2n$ soustractions. Cette méthode est donc très efficace, surtout lorsque nous n'avons besoin que de quelques valeurs propres (environ $4kp^2$ flops pour p valeurs propres), en comparaison avec la méthode QR qui prend environ $2kn^2$ flops pour k itérations.

Annexe A

Algorithme de la tridiagonalisation de Householder

Pour trouver A_{n-1} et les valeurs propres d'une matrice symétrique $\in \mathbf{R}^{n\times n}$.

Input:

n: ordre de $A = A_1$

tol: constante de la machine $= \frac{\beta^{L-1}}{\text{epsilon machine}} = m$

a: vecteur $n \times n$, données de A.

Output:

a

d: vecteur $n \times 1$: éléments de la diagonale de A_{n-1}

e: vecteur $n \times 1$: éléments de la co-diagonale de A_{n-1}
où $e[1]$ est défini comme 0.

$e2$: vecteur $n \times 1$: $e2[i] = (e[i])^2$

Procédure[1]

```
Begin integer i, j, k, l;
    real f, g, h;
    for i := 1 step 1 until n do
        d[i] := a[i, i];
    for i : n step −1 until 1 do
        Begin l := i − 1; h := 0;
            for k := 1 step 1 until l do
                h := h + a[i, k] × a[i, k];
            if h ≤ tol then
                Begin e[i] := e2[i] = 0; goto skip
                end;
            e2[i] := h; f := a[i, i − 1];
        e[i] := g := if ≥ 0 then −sqrt(h) else sqrt(h);
        h := h − f × g; a[i, i − 1] := f − g; f := 0;
        for j := 1 step 1 until l do
        Begin g := 0;
            for k := 1 step 1 until j do
                g := g + a[j, k] × a[i, k];
            for k := j + 1 step 1 until l do
                g := g + a[k, j] × a[i, k];
            g := e[j] := g/h; f := f + g × a[i, j];
        end j;
        h := f/(h + h);
        for j := 1 step 1 until l do
        Begin f := a[i, j]; g := e[j] := e[j] − h × f;
            for k := 1 step 1 until j do
```

[1] J. .H Wilkinson et C. Reinsch, *Linear Algebra*, vol. II, Springer-Verlag, Berlin, (1971), p. 203.

$a[j,k] := a[j,k] - f \times e[k] - g \times a[i,k]$
end j;
skip: $h := d[i]; d[i] := a[i,i]; a[i,i] := h$;
end i;
end Procédure.

Bibliographie

ARMINJON, Paul, *Analyse numérique matricielle*, Les Presses de l'Université de Montréal, Montréal, 1978, pp. 314–340.

GOLUB, Gene H. et VAN LOAN, Charles, *Matrix computations*, Johns Hopkins University Press, Baltimore, MD, 1st edition, 1983, pp. 276–283.

HAGER, William W., *Applied Numerical Linear Algebra*, Prentice Hall, Englewood Cliffs, NJ, 1988, pp. 70–77, 278–288.

PARLETT, Beresford N., *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980, pp. 111–138.

WILKINSON, J. H. et REINSCH, C., *Linear Algebra*, vol. II, Springer-Verlag, Berlin, 1971, pp. 192–222.

Chapitre 12

Forme de Jordan

Dans ce chapitre on présente une démonstration constructive de l'existence de la forme canonique de Jordan d'une matrice triangulaire inférieure quelconque obtenue au moyen de similarités produites par des transformations rationnelles sur les lignes et sur les colonnes. On illustre la méthode au moyen d'exemples simples. Chantal Boisvert et Martin Renaud ont construit l'exemple de dimension 8 et ont traduit l'article [9] en français. Elinor Friedman a réalisé la frappe de ce texte en AMS-TeX, qui par la suite a été mis en LaTeX.

12.1 Introduction

La démonstration traditionnelle de l'existence de la forme canonique de Jordan d'une matrice complexe quelconque M, que l'on retrouve, par exemple, dans Gantmacher [5] (vol. 1, ch. 7, pp. 175 ss.) est longue et compliquée. Ce fait est reconnu par Strang [12] (voir p. 228) où même une démonstration plus simple de Filippov [2] est reléguée à l'appendice, et dans Flechter–Sorensen [3], Galperin–Waksman [4], Brualdi [1], Hartl [8] et Lesky [11], où l'on décrit différentes simplifications.

L'objet de ce chapitre est de rendre le théorème et sa preuve accessibles au niveau d'un cours d'algèbre linéaire prégradué. On suppose que la lectrice et la lecteur qui voient le théorème pour la première fois, connaissent l'élimination gaussienne et son interprétation matricielle et savent l'appliquer sur les lignes et les colonnes afin de construire une matrice similaire à une matrice triangulaire donnée et dans la forme de Jordan prescrite. Cet article suit de près le travail de Howland–Vaillancourt [9], à l'exception de l'exemple de dimension 8 qui est nouveau.

On remarque que cette construction n'est pas conçue pour la programmation informatique — de fait, pour conserver la forme triangulaire de la matrice aucun pivot n'est employé. Il s'agit plutôt d'une preuve constructive que l'on peut tester en arithmétique rationnelle exacte sur de courts exemples bien que sa portée s'étende, du moins en principe, aux matrices réelles ou complexes d'ordre

quelconque.

La construction pratique de la forme de Jordan présentée dans la monographie de Golub–Van Loan [6], pp. 390–392, se base sur l'exposé de Golub–Wilkinson [7] et fait usage de la décomposition en valeurs singulières de la partie triangulaire stricte de chaque bloc diagonal aux valeurs propres identiques de la forme de Schur de M. Enfin, Kagström–Ruhe [10] ont mis en œuvre cette méthode pratique sur ordinateur.

12.2 De la forme de Schur à la forme de Jordan

On peut exprimer la forme inférieure de Jordan J d'une matrice $M \in \mathbf{C}^{n\times n}$ ou $\in \mathbf{R}^{n\times n}$ au moyen d'une similarité:

$$J = T^{-1}MT, \tag{12.1}$$

où les colonnes de la matrice de la transformation T proviennent des vecteurs caractéristiques et des vecteurs principaux de M, en supposant qu'une base complète de ces vecteurs existe et que l'on puisse la trouver. Pour construire J, et, si requis, T, on montre que J est reliée à la forme de Schur triangulaire inférieure S de M par une similarité triangulaire inférieure:

$$J = LSL^{-1}, \tag{12.2}$$

On sait que S s'obtient à partir de M par une similarité unitaire:

$$S = Q^*MQ. \tag{12.3}$$

Soit la décomposition QR de $(T^{-1})^*$:

$$(T^{-1})^* = QR,$$

où la notation anticipe le résultat et R est triangulaire supérieure. Si l'on substitue

$$T^{-1} = R^*Q^*, \qquad T = Q(R^*)^{-1}$$

dans (12.1), on obtient la forme de Jordan:

$$J = R^*Q^*MQ(R^*)^{-1}.$$

On voit que la forme de Schur:

$$S = Q^*MQ = (R^*)^{-1}JR^* \tag{12.4}$$

est triangulaire inférieure. Enfin, en notant $R^* = L$, une matrice triangulaire inférieure, (12.2) suit de (12.4).

Cette représentation suggère que l'on peut obtenir J à partir de S par une suite finie d'opérations élémentaires sur les lignes et les colonnes de S, à savoir, celles utilisées dans la solution d'un système linéaire par élimination et par des

procédures de décomposition triangulaire apparentée. On démontrera la véracité de cette conjecture.

La forme de Schur triangulaire inférieure S de M sera notre point de départ pour la construction de la forme de Jordan J. On suppose que S est disponible. En particulier, on suppose que les valeurs propres de M sont connues et que, dans la similarité (12.3), on a choisi la matrice Q de façon que les valeurs propres égales de M se suivent sur la diagonale de S. On peut donc partitionner S en sous-matrices S_{ij} où chaque bloc diagonal S_{ii} de dimension maximale est triangulaire inférieur et contient toutes les valeurs propres $\lambda = \lambda_i$ sur la diagonale. Ainsi, les éléments diagonaux de différentes sous-matrices principales sont différents. Chaque S_{ii} est de la forme $S_{ii} = \lambda_i I + N_{ii}$, où N_{ii} est triangulaire inférieure stricte et l'ordre de S_{ii} est égal à la multiplicité algébrique de la valeur propre associée λ_i.

12.3 Elimination sur les lignes et les colonnes

L'élimination sur les lignes se fait par multiplication à gauche par des matrices triangulaires inférieures élémentaires $L_{ij}(m)$. Dans le langage de l'élimination gaussienne, l'application de $L_{ij}(m)$ sur M multiplie la $j^{\text{ième}}$ ligne de M, qui contient le pivot m_{jk}, par le multiplicateur m pour éliminer l'élément m_{ik} de la $i^{\text{ième}}$ ligne de M — ordinairement pour annuler m_{ij}. L'élimination sur les colonnes se fait par multiplication à droite. Ainsi $ML_{ij}(m)$ représente une opération élémentaire sur la $j^{\text{ième}}$ colonne au moyen de la $i^{\text{ième}}$ colonne pivot que l'on multiplie par m. Voici un exemple où $M = S$ est triangulaire inférieure d'ordre 4, avec $i = 3$, $j = 2$:

$$
\begin{aligned}
&L_{32}(m)SL_{32}^{-1}(m) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{11} & 0 & 0 & 0 \\ s_{21} & s_{22} & 0 & 0 \\ s_{31} & s_{32} & s_{33} & 0 \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} s_{11} & 0 & 0 & 0 \\ s_{21} & s_{22} & 0 & 0 \\ s_{31} - ms_{21} & s_{32} - ms_{22} + ms_{33} & s_{33} & 0 \\ s_{41} & s_{42} + ms_{43} & s_{43} & s_{44} \end{bmatrix}.
\end{aligned}
$$

La similarité $L_{ij}SL_{ij}^{-1}$ — essentiellement une élimination gaussienne par lignes et par colonnes — peut être utilisée pour annuler un élément de la matrice de Schur triangulaire inférieure S. Cette transformation préserve la forme triangulaire inférieure et les zéros introduits aux étapes antérieures si l'on introduit des zéros successivement dans un ordre approprié. On obtient ainsi la forme de Jordan de M, J, à partir de la forme de Schur S par l'insertion systématique d'autant de zéros que possible au moyen:

(a) de similarités triangulaires inférieures: LSL^{-1}, où L est une matrice triangulaire inférieure,

(b) de similarités diagonales: DSD^{-1}, où D est une matrice diagonale, et

(c) de permutations cycliques de lignes et de colonnes:PSP^T, ou P est une matrice obtenue par une permutation des lignes ou colonnes de la matrice identité,

tout en préservant la forme inférieure de S. Par conséquent, dans l'exemple précédent, on peut éliminer l'élément $(3,2)$ de $L_{ij}SL_{ij}^{-1}$ avec le multiplicateur

$$m = \frac{s_{32}}{s_{22} - s_{33}} \tag{12.5}$$

si $s_{33} \neq s_{22}$.

Notation 12.1 *On note*

$$\{i_1, i_2, i+3, \ldots, i_k\} \mapsto \{i_k, i_1, i_2, \ldots, i_k - 1\}$$

la permutation cyclique des k lignes l_{i_j} et colonnes c_{i_j}, d'éfinie par les opérations associatives suivantes sur les lignes et sur les colonnes:

$$\begin{array}{lllllll} l_{i_2} & := & l_{i_1} & c_{i_2} & := & c_{i_1}, \\ l_{i_3} & := & l_{i_2} & c_{i_3} & := & c_{i_2}, \\ & \vdots & & & \vdots & \\ l_{i_1} & := & l_{i_k} & c_{i_1} & := & c_{i_k}. \end{array}$$

La permutation cyclique lignes-colonnes qu'on vient de définir, fait passer la dernière ligne d'un bloc à la première ligne du bloc et pousse les autres lignes d'un cran vers le bas, puis fait de même avec la dernière colonne en poussant les autres colonnes d'un cran vers la droite. L'élément $s_{i,i-1}$ de la codiagonale se retrouve un cran plus bas, c'est-à-dire

$$s_{i+1,i} := s_{i,i-1}.$$

12.4 Eléments de la diagonale distincts

Dans ce cas simple, les similarités par éliminations produisent le résultat bien connu qui suit.

Lemme 12.1 *Si les éléments de la diagonale de S, s_{ii}, sont tous différents, alors la forme de Jordan est la matrice diagonale avec ces mêmes éléments diagonaux.*

Démonstration. Puisque les éléments de la diagonale de S (les valeurs propres de M) sont tous différents, en choisissant le multiplicateur (12.5) approprié on a:

$$(L_{21}SL_{21}^{-1})_{21} = 0.$$

De même, on a

$$(L_{31}L_{21}SL_{21}^{-1}L_{31}^{-1})_{31} = 0$$

sans détruire le zéro déjà introduit. En effet, cette étape modifie la ligne 3 et la colonne 1 et, puisque $L_{21}SL_{21}^{-1}$ est triangulaire inférieure, l'élément de la colonne 3 ajouté à celui de la colonne 1 sur la $2^{\text{ième}}$ ligne est zéro. De même,

$$(L_{41}L_{31}L_{21}SL_{21}^{-1}L_{31}^{-1}L_{41}^{-1})_{41} = 0$$

sans détruire les zéros déjà introduits, etc. En procédant de la colonne 1 aux colonnes $2, 3, \ldots, (n-1)$, on peut réduire S à la forme diagonale D. C'est la forme de Jordan de M. Ses éléments diagonaux sont ceux de S. On représente le processus complet par l'expression:

$$LSL^{-1} = D,$$

où

$$L = L_{n\,n-1} \cdots L_{41}L_{31}L_{21}. \quad \square$$

12.5 Eléments de la diagonale identiques

Supposons maintenant que les éléments de la diagonale de S (les valeurs propres de M) ne sont pas tous différents de sorte que, dans la partition de S décrite ci-haut, certaines des sous-matrices S_{ii} sont de dimension supérieure à 1. En suivant la procédure exposée ci-haut dans le cas de valeurs propres distinctes, on peut successivent réduire à zéro les éléments des sous-matrices non diagonales S_{ij} de S. D'ailleurs, en procédant de la colonne 1 à la colonne n et de haut en bas dans chaque colonne, comme auparavant, les matrices diagonales S_{ii} resteront inchangées par ce processus. En ré-étiquetant ces dernières: $S_1, S_2, \ldots, S_m$, il est clair que l'on peut considérer chacune de ces sous-matrices séparément et indépendemment des autres. Il est convenable de considérer la matrice triangulaire inférieure stricte N dans chaque cas, puisque les similarités ne changent pas λI. Alors,

$$N = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ s_{21} & 0 & 0 & \cdots & 0 \\ s_{31} & s_{32} & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ s_{n1} & s_{n2} & s_{n3} & \cdots & 0 \end{bmatrix}.$$

Il faut remarquer que la première ligne et la dernière colonne sont nulles.

Les étapes d'élimination utilisées sur N sont plus simples que celles appliquées ci-haut: l'élimination sur les lignes ou les colonnes utilise les éléments de la codiagonale $s_{21}, s_{32}, \ldots, s_{n\,n-1}$ comme pivots, sans oublier, à chaque étape, de préserver la similarité en effectuant l'opération correspondante sur les colonnes ou les lignes.

Notation 12.2 *On note*

$$l_i := l_i - m\, l_j \qquad c_j := c_j + m\, c_i,$$

la similarité qui eélimine un élément de la ligne l_i par la ligne l_j et l'opération correspondante sur les colonnes c_j et c_i; on parlera d'élimination lignes-colonnes (ou colonnes-lignes).

Dans la suite on distinguera deux cas: celui où aucun élément de la codiagonale n'est nul et celui où au moins un élément de la codiagonale est nul.

12.6 Aucun zéro sur la codiagonale

Dans le cas où les pivots codiagonaux sont non nuls, les raisonnements qui suivent démontrent le lemme suivant.

Lemme 12.2 *Soit une matrice triangulaire inférieure d'ordre n:*

$$S = \lambda I + N,$$

où N est triangulaire inférieure stricte. Si les pivots codiagonaux de S sont non nuls:

$$s_{21} s_{32} \cdots s_{n,n-1} \neq 0,$$

alors la forme de Jordan de S consiste en un seul bloc de Jordan, c'est-à-dire, on a λ sur la diagonale et 1 sur la codiagonale.

Par exemple, on réduit le pivot s_{21} par la similarité diagonale suivante:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/s_{21} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ s_{21} & 0 & 0 & 0 \\ s_{31} & s_{32} & 0 & 0 \\ s_{41} & s_{42} & s_{43} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_{21} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ s_{31} & s_{32}s_{21} & 0 & 0 \\ s_{41} & s_{42}s_{21} & s_{43} & 0 \end{bmatrix}.$$

Il est utile de garder la notation s_{ij} pour les éléments qui ne sont pas des pivots de cette matrice réduite. On peut mettre à zéro les éléments $s_{31}, s_{41}, \ldots, s_{n1}$ de la première colonne par la soustraction de multiples de la ligne 2 des lignes suivantes. A la différence de la formule (5), les multiplicateurs de la présente méthode sont des éléments de la matrice elle-même. Pour préserver la similarité, on doit additionner à la colonne 2 des multiples des colonnes suivantes. Tous les zéros introduits par les opérations sur les lignes sont préservés aux étapes ultérieures ainsi que les pivots. De cette façon, il est possible de mettre $s_{42}, s_{52}, \ldots, s_{n2}$ à zéro. Après l'introduction de zéros dans les colonnes $1, 2, \ldots, (n-2)$, on obtient la matrice $J - \lambda I$, où J est la forme de Jordan de S. Donc, $J = L(N + \lambda I)L^{-1}$

où L est tiangulaire inférieure puisque c'est le produit de matrices diagonales élémentaires et de matrices triangulaires inférieures.

On illustre cette réduction au moyen d'une matrice d'ordre 4 aux pivots déjà réduits:

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 4 & 5 & 1 & 0 \end{bmatrix}.$$

Notons m le multiplicateur, l_i la $i^{\text{ième}}$ ligne et c_j la $j^{\text{ième}}$ colonne. On emploie trois similarités:

$$\begin{matrix} 0 & & & \\ 1 & 0 & & \\ 3 & 1 & 0 & \\ 4 & 5 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 3 \\ \longrightarrow \\ l_3 := l_3 - 3l_2 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 4 & 5 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 3 \\ \longrightarrow \\ c_2 := c_2 + 3c_3 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 4 & 8 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 4 & 8 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 4 \\ \longrightarrow \\ l_4 := l_4 - 4l_2 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 8 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 4 \\ \longrightarrow \\ c_2 := c_2 + 4c_4 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 8 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 8 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 8 \\ \longrightarrow \\ l_4 := l_4 - 8l_3 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 \end{matrix} \quad \begin{matrix} m = 8 \\ \longrightarrow \\ c_3 := c_3 + 8c_4 \end{matrix} \quad \begin{matrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 \end{matrix}$$

Dans ce cas, l'élimination sur l_i produit le zéro nécessaire. Puisque le pivot est sous la diagonale, l'élimination associée sur c_j ne détruit pas le zéro que l'on vient d'introduire ni les zéros obtenus aux étapes antérieures.

On peut réduire ainsi des matrices N d'ordre quelconque. De plus, on peut écrire la forme de Jordan dès que l'on voit qu'aucun des éléments de la codiagonale n'est nul, c'est-à-dire, le polynôme minimal d'une matrice de Schur S est son polynôme caractéristique si les éléments de la codiagonale correspondant à chacune des valeurs propres sont non nuls.

12.7 Des zéros sur la codiagonale

Lorsqu'il y a un ou plusieurs zéros sur la codiagonale de N, il est utile d'utiliser une induction sur l'ordre n de N pour construire la forme de Jordan, et ainsi en démontrer l'existence.

Dans ce cas, les raisonnements qui suivent démontrent le lemme suivant.

Lemme 12.3 *Soient une matrice triangulaire inférieure d'ordre $n+1$:*

$$S = \lambda I + N,$$

et la sous-matrice triangulaire inférieure stricte N de la forme

$$N = \begin{bmatrix} \tilde{N} & 0 & 0 \\ 0 & \tilde{E} & 0 \\ u^T & e^T & 0 \end{bmatrix} \quad \begin{matrix} n-k \\ k \\ 1 \end{matrix}$$

où la sous-matrice principale d'ordre n

$$\begin{bmatrix} \tilde{N} & 0 \\ 0 & \tilde{E} \end{bmatrix}$$

est déjà dans une forme de Jordan et $\tilde{E}$ est son plus grand bloc de Jordan. Alors on peut transformer N en sa forme de Jordan par les transformations rationnelles considérées dans ce travail.

Il faut se souvenir que la première ligne et la dernière colonne de N sont nulles, quelqu'en soit l'ordre; alors on peut faire l'élimination sur les lignes par la dernière ligne et sur les colonnes par la première colonne sans se soucier des opérations sur les colonnes et sur les lignes respectivement requises pour préserver la similarité, puisque ces dernières sont sans effet aucun. Ceci fut illustré aux étapes 2 et 3 de l'exemple précédent.

Considérons le premier cas non trivial, d'ordre 3:

$$N = \begin{bmatrix} 0 & 0 & 0 \\ a & 0 & 0 \\ b & c & 0 \end{bmatrix} \qquad \text{et} \quad ac = 0.$$

Il y a trois situations possibles:

$$a \neq 0, \quad c = 0: \quad \begin{bmatrix} 0 & 0 & 0 \\ a & 0 & 0 \\ b & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ b & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$a = 0, \quad c \neq 0: \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b & c & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{b}{c} & 1 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$a = c = 0, \quad b \neq 0: \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ b & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ b & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

On a noté $\sim$ une similarité par des opérations obvies sur les lignes et les colonnes. Il est clair que l'on doit utiliser des similarités diagonales, des permutations cycliques lignes-colonnes et une élimination sur les lignes ou sur les colonnes respectivement au moyen de la dernière ligne ou de la première colonne.

Considérons maintenant le cas suivant, d'ordre 4, où l'on a déjà transformé la première sous-matrice principale d'ordre 3 dans sa forme de Jordan et que

l'on a placé le plus grand bloc de Jordan dans le coin inférieure droit au moyen de permutations cycliques lignes-colonnes.

De nouveau, il y a trois cas possibles:

Cas (a):

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a & b & c & 0 \end{bmatrix}; \qquad |a| + |b| + |c| \neq 0. \tag{12.6}$$

Par des échanges lignes-colonnes, à l'exclusion de la dernière ligne et de la dernière colonne, sans perte, on peut supposer que $c \neq 0$. Alors,

$$N \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{a}{c} & \frac{b}{c} & 1 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

car l'élimination sur les colonnes au moyen de la $3^{\text{ième}}$ colonne est suffisante et l'opération correspondante sur les lignes, requise pour préserver la similarité, est sans effet aucun.

Cas (b):

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & c & 0 \end{bmatrix}; \qquad a, b, c \text{ quelconques.} \tag{12.7}$$

$$N \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & c & 0 \end{bmatrix} \quad \text{par élimination au moyen de la dernière ligne, si } b \neq 0.$$

Si $c \neq 0$:

$$N \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{a}{c} & 0 & 1 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

par élimination au moyen de la première colonne.

Si $c = 0$ et $a \neq 0$, on voit que $a \neq 0$ est bien placé pour une élimination de ligne ou de colonne mais les pivots nécessaires sont nuls. Dans ce cas, pour placer a sur la codiagonale et en faire un pivot on emploie la permutation cyclique lignes-colonnes $\{2, 3, 4\} \mapsto \{4, 2, 3\}$:

$$N \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Cas (c):

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & c & 0 \end{bmatrix}; \qquad a, b, c \quad \text{quelconques.} \tag{12.8}$$

$$N \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \delta & 0 \end{bmatrix} \qquad \begin{array}{l} \delta = 0 \quad \text{si} \quad c = 0, \\ \delta = 1 \quad \text{si} \quad c \neq 0. \end{array}$$

Notons que l'on peut utiliser les stratégies employées dans les cas (12.6) et (12.8) sur des matrices N d'ordre quelconque dans des ciconstances similaires, c'est-à-dire lorsque les pivots dans la première sous-matrice principale sont tous égaux.

La stratégie utilisée dans le cas (12.7) lorsque l'élément $(4, 3)$ est non nul peut être aussi utilisée pour des matrices d'ordre $n+1$. Ceci implique des éliminations sur les lignes pour mettre des zéros dans la ligne $n+1$ sous les pivots (tous égaux à 1) des lignes 2 à n, puis des éliminations correspondantes sur les colonnes pour mettre à zéro les éléments qui restent à gauche dans la dernière ligne. De plus, des éliminations supplémentaires sur les colonnes peuvent être nécessaires pour restaurer la forme de Jordan.

D'une façon plus précise, on obtient la forme:

$$N \sim \begin{bmatrix} \tilde{N} & 0 & 0 \\ 0 & \tilde{E} & 0 \\ u^T & e^T & 0 \end{bmatrix}, \qquad (n+1) \times (n+1),$$

où $\tilde{E}$ est le plus grand bloc de Jordan, d'ordre k, dans la sous-matrice principale, $\tilde{N}$ est le reste, d'ordre $(n-k)\times(n-k)$, de la sous-matrice principale, fait de blocs de Jordan d'ordre $\leq k$, e^T est le vecteur transposé $(0, 0, \ldots, 0, 1)$ d'ordre $1 \times k$ et u^T est quelconque, à l'exception des éléments nuls sous les pivots non nuls de $\tilde{N}$. On peut utiliser l'élimination sur les colonnes pour mettre à zéro tout élément non nul de u^T; les opérations correspondantes sur les lignes introduiront des éléments non nuls dans la ligne n, en haut et à gauche des éléments initialement non nuls de u^T.

Puisque

$$E = \begin{bmatrix} \tilde{E} & 0 \\ e^T & 0 \end{bmatrix}$$

est une forme de Jordan d'ordre $k + 1$, on voit que

$$N \sim \begin{bmatrix} \tilde{N} & 0 \\ e_{k+1}u^T & E \end{bmatrix}$$

où e_j est la $j^{\text{ième}}$ colonne de la matrice identité d'ordre $k + 1$. Alors, on obtient par élimination sur les colonnes:

$$N \sim \begin{bmatrix} I & 0 \\ e_k u^T & I \end{bmatrix} \begin{bmatrix} \tilde{N} & 0 \\ e_{k+1}u^T & E \end{bmatrix} \begin{bmatrix} I & 0 \\ -e_k u^T & I \end{bmatrix} = \begin{bmatrix} \tilde{N} & 0 \\ e_k u^T \tilde{N} & E \end{bmatrix}$$

puisque les opérations sur les lignes pour préserver la similarité déplaceront les éléments non nuls de u^T à la ligne n une colonne à gauche si le pivot de cette colonne est non nul. Ceci donne $u^T\tilde{N}$ dans la ligne n. On peut faire d'autres éliminations sur les colonnes, de ligne à ligne, tant que l'on dispose de pivots non nuls, c'est-à-dire pour toutes les lignes de E. La dernière élimination sur les lignes, dans la $2^{\text{ième}}$ ligne de E, produira la matrice

$$N \sim \begin{bmatrix} \tilde{N} & 0 \\ e_1 u^T \tilde{N}^k & E \end{bmatrix}.$$

Mais, étant donné que les blocs de Jordan de $\tilde{N}$ sont tous d'ordre $\leq k$, alors $\tilde{N}^k = 0$. Ceci est, en fin de compte, la forme de Jordan. Le déplacement des éléments non nuls de la dernière ligne vers le haut et vers la gauche doit nécessairement se terminer avant l'épuisement de la réserve de pivots dans E.

La stratégie appliquée dans le cas (7) lorque l'élément $(4,3)$ est nul, est aussi applicable à des matrices d'ordre quelconque. Ceci implique des éliminations sur les lignes pour introduire autant de zéros que possible dans la dernière ligne et l'identification du premier élément non nul dans la dernière ligne en allant de la droite vers la gauche. Cet élément sera nécessairement à gauche du plus grand bloc de Jordan et sous un pivot nul. Les colonnes à droite de cet élément étant sous forme de Jordan, on peut placer cet élément non nul par une permutation cyclique lignes-colonnes dans une position pivotale (codiagonale) tout en laissant les colonnes à droite en forme de Jordan. Il faut donc considérer une sous-matrice principale d'ordre m, $m < n$, dont l'élément $(m, m-1)$ est non nul. Par l'hypothèse d'induction, on peut transformer cette matrice en une forme de Jordan et donc transformer toute la matrice en une forme de Jordan.

Donc, dans le cas $n = 5$ et $b \neq 0$, la permutation cyclique lignes-colonnes: $\{3,4,5\} \mapsto \{4,5,3\}$, et une élimination colonnes-lignes produisent les matrices similaires:

$$N = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ a & b & 0 & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ a & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

On peut donc toujours obtenir la forme de Jordan d'une matrice strictement triangulaire inférieure N d'ordre $(n+1)$ à partir de sa sous-matrice principale d'ordre n. On obtient ce résultat par induction pour une matrice N d'ordre quelconque. Ce résultat, combiné avec les autres cas considérés ci-haut, démontre le théorème suivant.

Théorème 12.1 *On peut transformer une matrice de Schur de dimension quelconque S en sa forme de Jordan J par des opérations rationnelles.*

Si l'on n'a pas besoin de la matrice qui transforme S en J, on peut trouver la forme de Jordan, dans un cas donné, en identifiant les sous-matrices principales

$S_1, \ldots, S_m$ de la forme de Schur S et en mettant à zéro toutes les sous-matrices non principales. Alors, on a immédiatement la forme de Jordan de chacune des sous-matrices principales dont aucun des éléments de la codiagonale n'est nul. *Il ne faut faire des calculs que pour les S_j qui ont au moins un élément codiagonal nul.* Dans chacun de ces cas, la forme de Jordan s'obtient en transformant les sous-matrices principales dans l'ordre croissant de leur dimension au moyen des stratégies élaborées ci-haut.

12.8 Un exemple d'ordre huit

Voici un premier exemple de matrice $M = M_1$, d'ordre 8, qui illustre la procédure décrite plus haut à l'aide d'opérations arithmétiques simples.

La sous-matrice principale 2×2 de M_1 est déjà sous la forme de Jordan. Alors, le premier cas à considérer est la sous-matrice principale 3×3. L'élimination sur les lignes suffit dans ce cas. Par conséquent, l'élimination lignes-colonnes:

$$l_3 := l_3 - l_2 \qquad c_2 := c_2 + c_3$$

(lire: la ligne 3 est remplacée par la ligne 3 moins la ligne 2, etc.) produit la matrice similaire M_2:

$$M_1 = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 1 & 1 & 0 & & & & & \\ 1 & 0 & 0 & 0 & & & & \\ 0 & 0 & 1 & 1 & 0 & & & \\ 1 & 1 & 0 & 1 & 0 & 0 & & \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 1 & 1 & 0 & & & \\ 1 & 1 & 0 & 1 & 0 & 0 & & \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{matrix} = M_2.$$

On considère la sous-matrice principale 4×4 de M_2. L'élimination lignes-colonnes:

$$l_4 := l_4 - l_2 \qquad c_2 := c_2 + c_4$$

produit la matrice similaire M_3.

On considère la sous-matrice principale 5×5 de M_3. Il faut d'abord permuter cycliquement les lignes et les colonnes $1, 2, 3, 4$ afin de placer le plus gros bloc de Jordan obtenu à date dans le coin inférieur droit. La permutaion cyclique lignes-colonnes:

$$\{1, 2, 3, 4\} \mapsto \{4, 1, 2, 3\}$$

produit alors la matrice similaire M_4:

$$M_3 = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 2 & 1 & 1 & 0 & & & \\ 1 & 2 & 0 & 1 & 0 & 0 & & \\ 0 & 2 & 1 & 1 & 0 & 1 & 0 & \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 1 & 0 & 2 & 1 & 0 & & & \\ 1 & 1 & 2 & 0 & 0 & 0 & & \\ 1 & 0 & 2 & 1 & 0 & 1 & 0 & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{matrix} = M_4.$$

L'élimination lignes-colonnes:

$$l_5 := l_5 - 2l_4 \qquad c_4 := c_4 + 2c_5$$

produit la matrice similaire M_5.

L'élimination sur les colonnes de M_5 est ensuite utilisée pour introduire un zéro en position $(5, 1)$. L'opération correspondante sur les lignes:

$$c_1 := c_1 - c_4 \qquad l_4 := l_4 + c_1$$

ne change rien. On obtient donc la matrice similaire M_6:

$$M_5 = \begin{matrix} 0 & & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 1 & 0 & 0 & 1 & 0 & & & \\ 1 & 1 & 2 & 0 & 0 & 0 & & \\ 1 & 0 & 2 & 1 & 0 & 1 & 0 & \\ 0 & 1 & 1 & 3 & 1 & 1 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 1 & 1 & 2 & 0 & 0 & 0 & & \\ 0 & 0 & 2 & 1 & 0 & 1 & 0 & \\ -3 & 1 & 1 & 3 & 1 & 1 & 1 & 0 \end{matrix} = M_6.$$

Considérons la sous-matrice 6×6 de M_6. Les deux éléments de la ligne 6 qui ont des pivots non nuls, sont réduits à zéro par les éléminations lignes-colonnes:

$$\begin{matrix} l_6 := l_6 - 2l_4 & \qquad & c_4 := c_4 + 2c_6 \\ l_6 := l_6 - \ l_3 & \qquad & c_3 := c_3 + \ c_6 \end{matrix}$$

pour produire la matrice similaire M_7.

L'élément en position $(6, 1)$ de M_7 ne peut être réduit à zéro car ses pivots de ligne et de colonne sont nuls. On effectue la permutation cyclique lignes-colonnes:

$$\{2, 3, 4, 5, 6\} \mapsto \{6, 2, 3, 4, 5\}$$

pour obtenir la matrice similaire M_8:

$$M_7 = \begin{matrix} 0 & & & & & & & \\ 0 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 3 & 3 & 0 & 1 & 0 & \\ -3 & 1 & 2 & 5 & 1 & 1 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 0 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 1 & 0 & 3 & 3 & 0 & 0 & \\ -3 & 1 & 1 & 2 & 5 & 1 & 1 & 0 \end{matrix} = M_8.$$

Les éléments en position $(7,4)$ et $(7,5)$ de la sous-matrice principale 7×7 de M_8 peuvent être réduits à zéro par les éléminations lignes-colonnes:

$$\begin{array}{ll} l_7 := l_7 - 3l_6 & c_6 := c_6 + 3c_7 \\ l_7 := l_7 - 3l_5 & c_5 := c_5 + 3c_7 \end{array}$$

pour produire la matrice similaire M_9.

Puisque l'élément de M_9 en position $(7,2)$ ne peut être annulé, faute de pivots, la permutation cyclique lignes-colonnes:

$$\{3,4,5,6,7\} \mapsto \{7,3,4,5,6\}$$

le placera en position pivot $(3,3)$ et produira la matrice similaire M_{10}:

$$M_9 = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 0 & 0 & & & & & \\ 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \\ -3 & 1 & 1 & 2 & 8 & 4 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ -3 & 1 & 1 & 1 & 2 & 8 & 4 & 0 \end{matrix} = M_{10}.$$

Les éliminations lignes-colonnes (les opérations sur les colonnes étant sans effet aucun):

$$\begin{array}{lcllcl} l_8 & := & l_8 - 8l_7 & c_7 & := & c_7 + 8c_8 \\ l_8 & := & l_8 - 2l_6 & c_6 & := & c_6 + 2c_8 \\ l_8 & := & l_8 - l_5 & c_5 & := & c_5 + c_8 \\ l_8 & := & l_8 - l_3 & c_3 & := & c_3 + c_8 \\ l_8 & := & l_8 + 3l_2 & c_2 & := & c_2 - 3c_8 \end{array}$$

produit la matrice similaire M_{11}.

L'élimination colonnes-lignes:

$$c_3 := c_3 - \frac{1}{4}c_7 \qquad l_7 := l_7 + \frac{1}{4}l_3$$

produit la matrice similaire M_{12}:

$$M_{11} = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 4 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{matrix} = M_{12}.$$

Puisque une opération sur les lignes d'abord annulerait l'étape précédente, on commence par les colonnes. L'élimination colonnes-lignes:

$$c_2 := c_2 - \frac{1}{4}c_6 \qquad l_6 := l_6 + \frac{1}{4}l_2$$

fait monter l'élément $(7, 2)$ en position $(6, 1)$ et produit la matrice similaire M_{13}.

Une dernière élimination colonnes-lignes (l'opération sur les lignes étant sans effet),

$$c_1 := c_1 - \frac{1}{4}c_5 \qquad l_5 := l_5 + \frac{1}{4}l_1$$

élimine l'élément $(6, 1)$ et produit la matrice similaire M_{14}:

$$M_{13} = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ \frac{1}{4} & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{matrix} = M_{14}.$$

Une division de la ligne 8 par 4 et une multiplication de la colonne 8 par 4 (sans effet):

$$l_8 := \frac{1}{4}l_8 \qquad c_8 := 4c_8$$

produisent la matrice similaire J qui est la forme de Jordan de M:

$$M_{14} = \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & \\ 1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} = J.$$

12.9 Un exemple d'ordre dix

Voici un second exemple de matrice $M = M_1$ d'ordre 10 qui illustre la procédure décrite plus haut à l'aide d'opérations arithmétiques simples.

La sous-matrice principale 2×2 de M_1 est déjà sous la forme de Jordan. Alors, le premier cas à considérer est la sous-matrice principale 3×3. L'élimination sur les lignes suffit dans ce cas. Par conséquent, l'élimination lignes-colonnes:

$$l_3 := l_3 - l_2 \qquad c_2 := c_2 + c_3$$

(lire: la ligne 3 est remplacée par la ligne 3 moins la ligne 2, etc.) produit la matrice similaire M_2:

$$M_1 = \begin{matrix} 0 \\ 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 \\ 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} = M_2.$$

Pour passer à la sous-matrice principale 4×4 de M_2, il faut d'abord permuter cycliquement les lignes et les colonnes $1, 2, 3$ pour placer le plus gros bloc de Jordan obtenu à date dans le coin inférieur droit. La permutaion cyclique lignes-colonnes:

$$\{1, 2, 3\} \mapsto \{3, 1, 2\}$$

produit la matrice similaire M_3.

L'élimination colonnes-lignes:

$$c_1 := c_1 - c_3 \qquad l_3 := l_3 + l_1$$

produit la matrice similaire M_4:

$$M_3 = \begin{matrix} 0 \\ 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 \\ 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} = M_4.$$

La sous-matrice principale 4×4 de M_4 est déjà sous la forme de Jordan. L'élimination lignes-colonnes:

$$l_5 := l_5 - l_4 \qquad c_4 := c_4 + c_5$$

produit la matrice similaire M_5.

L'élément -1 de M_5 en position $(5,1)$ ne peut être réduit à zéro car ses pivots de ligne et de colonne sont nuls. On effectue la permutation cyclique lignes-colonnes:

$$\{2,3,4,5\} \mapsto \{5,2,3,4\}$$

et une similarité diagonale pour obtenir la matrice similaire M_6:

$$M_5 = \begin{matrix} 0 \\ 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 \\ 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix} = M_6.$$

La sous-matrice principale 5×5 de M_6 est déjà sous la forme de Jordan. Alors les trois pivots non nuls de la ligne 6 sont mis à zéro par les éliminations lignes-colonnes:

$$\begin{array}{ll} l_6 := l_6 - l_5 & c_5 := c_5 + c_6 \\ l_6 := l_6 - l_4 & c_4 := c_4 + c_6 \\ l_6 := l_6 + l_2 & c_2 := c_2 - c_6 \end{array}$$

et nous obtenons la matrice similaire M_7.

La sous-matrice principale 6×6 de M_7 est déjà sous la forme de Jordan. Trois des éléments non pivots de la ligne 7 sont mis à zéro par les éliminations lignes-colonnes:

$$\begin{array}{ll} l_7 := l_7 - 2l_6 & c_6 := c_6 + 2c_7 \\ l_7 := l_7 - 2l_5 & c_5 := c_5 + 2c_7 \\ l_7 := l_7 + \ l_2 & c_2 := c_2 - \ c_7 \end{array}$$

pour produire la matrice similaire M_8:

$$M_7 = \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 1 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ -1 & -1 & 0 & 2 & 2 & 1 & 0 & & & \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & & \\ -1 & -1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array} \sim \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 1 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & & \\ -1 & -2 & 0 & 1 & 3 & 2 & 1 & 1 & 0 & \\ 0 & -2 & 1 & 0 & 3 & 2 & 1 & 0 & 1 & 0 \end{array} = M_8.$$

L'élément -1 en position $(7,2)$ de M_8 est mis à zéro par l'élimination colonnes-lignes:

$$c_2 := c_2 + c_6 \qquad l_6 := l_6 - l_2,$$

mais cette dernière introduit -1 en position $(6,1)$; on le réduit à zéro par l'élimination colonnes-lignes:

$$c_1 := c_1 + c_5 \qquad l_5 := l_5 - l_1$$

ce qui produit la matrice similaire M_9.

Trois des éléments non pivots de la ligne 8 de M_9 sont mis à zéro par les éliminations lignes-colonnes:

$$\begin{array}{ll} l_8 := l_8 - l_6 & c_6 := c_6 + c_8 \\ l_8 := l_8 - l_4 & c_4 := c_4 + c_8 \\ l_8 := l_8 - l_2 & c_2 := c_2 + c_8 \end{array}$$

L'élément -1 en position $(8,2)$ ne peut être réduit à zéro car ses pivots de ligne et de colonne sont nuls. On effectue la permutation cyclique lignes-colonnes:

$$\{3,4,5,6,7,8\} \mapsto \{8,3,4,5,6,7\}$$

et une similarité diagonale pour obtenir la matrice similaire M_{10}:

$$M_9 = \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 0 & 0 & & & & & & & \\ 0 & 0 & 1 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 1 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & & \\ 2 & 0 & 0 & 2 & 3 & 2 & 1 & 1 & 0 & \\ 3 & 0 & 1 & 0 & 3 & 2 & 1 & 0 & 1 & 0 \end{array} \sim \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & \\ 2 & -1 & 1 & 0 & 2 & 3 & 3 & 1 & 0 & \\ 3 & 0 & 0 & 1 & 0 & 3 & 2 & 1 & 1 & 0 \end{array} = M_{10}.$$

Les éliminations lignes-colonnes:

$$\begin{array}{ll} l_9 := l_9 - 3l_8 & c_8 := c_8 + 3c_9 \\ l_9 := l_9 - 3l_7 & c_7 := c_7 + 3c_9 \\ l_9 := l_9 - 2l_6 & c_6 := c_6 + 2c_9 \\ l_9 := l_9 + \ l_3 & c_3 := c_3 - \ c_9 \\ l_9 := l_9 - 2l_2 & c_2 := c_2 + 2c_9 \end{array}$$

produit la matrice similaire M_{11}.

L'élément 1 en position $(9,3)$ de M_{11} peut être mis à zéro par l'élimination colonne:

$$c_3 := c_3 - c_8$$

mais l'opération ligne correspondante:

$$l_8 := l_8 + l_3$$

introduit 1 à la position $(8,2)$. Cet élément ne peut être mis à zéro par une opération ligne puique l'on reviendrait à la similarité antérieure. Alors on applique l'élimination colonne:

$$c_2 := c_2 - c_7$$

mais l'opération ligne correspondante:

$$l_7 := l_7 + l_2$$

introduit 1 en position $(7,1)$. On réduit cet élément à zéro par l'élimination colonnes-lignes:

$$c_1 := c_1 - c_6 \qquad l_6 := l_6 + l_1$$

(l'opération ligne étant sans effet). On a donc la matrice similaire M_{12}:

$$M_{11} = \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 3 & 2 & -1 & 1 & 0 & 5 & 5 & 4 & 1 & 0 \end{array} \sim \begin{array}{cccccccccc} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ -2 & -3 & -5 & 1 & 0 & 5 & 5 & 4 & 1 & 0 \end{array} = M_{12}.$$

Cette dernière est clairement similaire à M_{13} par élimination ligne si l'on se rappelle que la colonne 10 est nulle.

L'élimination colonnes-lignes:

$$c_3 := c_3 + 5c_9 \qquad l_9 := l_9 - 5l_3$$

déplace l'élément non nul de la position $(10, 3)$ de M_{13} à la position $(9, 2)$. L'élimination colonnes-lignes:

$$c_2 := c_2 + 5c_8 \qquad l_8 := l_8 - 5l_2$$

le déplace de la position $(9, 2)$ à la position $(8, 1)$. Finalement, l'élimination colonnes-lignes:

$$c_1 := c_1 + 5c_7 \qquad l_7 := l_7 - 5l_1$$

le met à zéro.

Ceci termine la réduction à la forme de Jordan de M:

$$M_{13} = \begin{matrix} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & -5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} \sim \begin{matrix} 0 & & & & & & & & & \\ 1 & 0 & & & & & & & & \\ 0 & 1 & 0 & & & & & & & \\ 0 & 0 & 0 & 0 & & & & & & \\ 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{matrix} = J.$$

Références

1. R. A. Brualdi, The Jordan canonical form: an old proof, *Amer. Math. Monthly* **94**(3) (1987), 257–267.

2. A. F. Filippov, A short proof of the theorem on reduction of a matrix to Jordan form, *Vestnik*, Moscow Univ., no 2 (1971), 18–19.

3. R. Flechter et D. C. Sorensen, An algorithmic derivation of the Jordan canonical form, *Amer. Math. Monthly* **90**(1) (1984), 12–16.

4. A. Galperin et Z. Waksman, An elementary approach to Jordan theory, *Amer. Math. Monthly* **87**(9) (1980), 728–732.

5. F. R. Gantmacher, *Théorie des matrices*, 2 tomes, trad. du russe par Ch. Sarthou, Dunod, Paris, 1966.

6. G. H. Golub et C. F. Van Loan, *Matrix Computation*, 2e éd., The Johns Hopkins University Press, Baltimore et London, 1989.

7. G. H. Golub et J. H. Wilkinson, Ill-conditioned eigensystems and the computation of the Jordan canonical form, *SIAM Review* **18**(4) (1976), 578–619.

8. J. Hartl, Ein einfacher Induktionsbeweis für die Existenz der Jordan-Normalform, *Arch. Math.* **50**(4) (1988), 323–327.

9. J. L. Howland et R. Vaillancourt, Rational transformation from Schur to Jordan form, *Computers Math. Applic.* **23**(4) (1992), 91–101.

10. B. Kagström et A. Ruhe, An algorithm for numerical computation of the Jordan normal form of a complex matrix, *ACM Trans. Math. Software* **6**(3) (1980), 398–419.

11. P. Lesky, Über die Konstruction der Jardanschen Normalform, *Elemente der Mathematik* **40**(5) (1985), 105–109.

12. G. Strang, *Linear Algebra and Its Applications*, 2e éd., Academic Press, New York, San Francisco, London, 1980.