

## Chapitre 9

## VALEURS PROPRES, VECTEURS PROPRES

## 1 Introduction

Du point de vue théorique, le problème de la détermination des vecteurs propres et valeurs propres d'une matrice est presque aussi bien connu que la résolution des systèmes linéaires. La physique fournit un grand nombre de problèmes équivalents à la diagonalisation d'une matrice symétrique ou hermitique. Je me limiterai essentiellement au cas des matrices réelles symétriques. Rappelons quelques propriétés des éléments propres d'une telle matrice.

J'appelle valeurs propres de la matrice carrée symétrique  $\mathbf{A}$  les solutions de l'équation en  $\lambda$  :

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0$$

si  $\mathbf{I}$  est la matrice identité. Si  $\mathbf{A}$  et  $\mathbf{I}$  sont de taille  $n \times n$ , les valeurs propres sont donc solutions de l'équation polynomiale de degré  $n$  :

$$(-1)^n(\lambda^n - C_{n-1}\lambda^{n-1} + \dots - C_1\lambda - C_0) = 0$$

appelée équation (ou polynôme) caractéristique. Dans le cas d'une matrice symétrique  $n \times n$ , l'équation (2) admet  $n$  racines réelles, distinctes ou non. Un certain nombre de méthodes anciennes (Leverrier, Krylov) forment le polynôme caractéristique et déterminent ses zéros. Ces méthodes sont abandonnées, car, en l'absence d'hypothèses supplémentaires sur  $\mathbf{A}$ , elles sont lentes (calcul des  $C_i$ ) et instables (recherche des racines). Par contre, ces deux opérations sont rapides pour une matrice tridiagonale. Vous verrez que l'une des méthodes les plus performantes consiste à «tridiagonaliser»  $\mathbf{A}$  en conservant ses éléments propres, puis à calculer les racines de son polynôme caractéristique.

Étant donné une matrice  $\mathbf{M}$  orthogonale, je dis que la matrice  $\mathbf{A}'$  est semblable à  $\mathbf{A}$  si  $\mathbf{A}' = \mathbf{M}^T \mathbf{A} \mathbf{M}$ . On a le théorème :  $\mathbf{A}'$  et  $\mathbf{A}$  ont mêmes valeurs propres.

Le vecteur propre  $\mathbf{u}$  associé à la valeur propre  $\lambda$  est défini par la relation :

$$\mathbf{A} \mathbf{u} = \lambda \mathbf{u}$$

Dans le cas particulier d'une matrice symétrique, les vecteurs propres (au nombre de  $n$ ) peuvent être choisis comme orthonormés : ils forment une base complète. Soit  $\mathbf{U}$  la matrice dont les colonnes sont les vecteurs propres. Les lectrices vérifieront sans peine que l'ensemble des  $n$  équations du type (3) peut s'écrire :

$$\mathbf{A} \mathbf{U} = \mathbf{U} \mathbf{\Lambda}$$

si  $\mathbf{\Lambda}$  désigne une matrice diagonale dont les seuls éléments non nuls sont les valeurs propres. De par sa définition,  $\mathbf{U}$  est orthogonale,  $\mathbf{U}^T = \mathbf{U}^{-1}$ . En effet, l'élément  $i, j$  du produit  $\mathbf{U}^T \mathbf{U}$  est le produit scalaire de la ligne  $i$  de  $\mathbf{U}^T$ , soit le vecteur ligne  $\mathbf{u}_i^T$ , par la colonne  $j$  de  $\mathbf{U}$ , soit  $\mathbf{u}^j$ . Comme les vecteurs propres sont orthonormés, j'ai  $(\mathbf{U}^T \mathbf{U})_{i,j} = \delta_{i,j}$  ; autrement dit, ce produit de matrices se réduit à la matrice unité. Finalement :

$$\mathbf{\Lambda} = \mathbf{U}^T \mathbf{A} \mathbf{U}$$

ce qui traduit le théorème suivant. Une matrice symétrique peut toujours être diagonalisée au moyen d'une similitude ; la matrice de transformation est orthogonale et ses colonnes forment un jeu complet de vecteurs propres.

## 2 Méthode de la puissance n-ième et méthodes dérivées

### 2.1 Puissance n-ième

Il arrive parfois que l'on recherche uniquement la plus grande des valeurs propres. La méthode de la puissance n-ième répond bien à ce besoin. L'algorithme est très simple. Ayant fait choix d'un vecteur initial  $\mathbf{x}^{(0)}$ , je forme par récurrence :

$$\mathbf{x}^{(n)} = \mathbf{A}\mathbf{x}^{(n-1)} = \mathbf{A}^n\mathbf{x}^{(0)}.$$

Le vecteur  $\mathbf{x}^{(n)}$  tend vers le vecteur propre associé à la plus grande valeur propre (en module); appelons-la  $\lambda_1$ . Le rapport de deux composantes homologues de  $\mathbf{x}^{(n)}$  et  $\mathbf{x}^{(n-1)}$ , soit  $x_k^{(n)}/x_k^{(n-1)}$ , tend vers  $\lambda_1$ . La démonstration est facile dans le cas d'une matrice symétrique, bien que la méthode qui vient d'être exposée soit plus générale. Soient  $\mathbf{u}^{(i)}$ ,  $\lambda_i$  les éléments propres de  $\mathbf{A}$ . Je peux développer  $\mathbf{x}^{(0)}$  sur la base des  $\mathbf{u}^{(i)}$  :

$$\mathbf{x}^{(0)} = \sum_i c_i \mathbf{u}^{(i)}$$

Appliquons  $\mathbf{A}$  :

$$\mathbf{x}^{(1)} = \mathbf{A}\mathbf{x}^{(0)} = \sum_i c_i \mathbf{A}\mathbf{u}^{(i)} = \sum_i c_i \lambda_i \mathbf{u}^{(i)}$$

puis  $\mathbf{A}^2$  :

$$\mathbf{x}^{(2)} = \mathbf{A}\mathbf{x}^{(1)} = \sum_i c_i \lambda_i^2 \mathbf{u}^{(i)}.$$

Il est facile d'écrire le cas général si l'on se souvient de ce que  $\mathbf{A}^k$  admet les mêmes vecteurs propres que  $\mathbf{A}$  et les  $\lambda_i^k$  comme valeurs propres :

$$\mathbf{x}^{(k)} = \sum_i c_i \lambda_i^k \mathbf{u}^{(i)} = \lambda_1^k c_1 \mathbf{u}^{(1)} + \lambda_1^{(k)} \sum_{i=2} c_i (\lambda_i/\lambda_1)^k \mathbf{u}^{(i)}.$$

Comme  $\lambda_i/\lambda_1 < 1$ , la propriété est démontrée.

*Exemple.* Cherchons la plus grande valeur propre de

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

à partir du vecteur  $[1, 0]^T$ . La suite des vecteurs itérés et les valeurs approchées de la valeur propre, calculées à partir de la deuxième composante, sont :

$k$	1	2	3	4	5	6
$\mathbf{x}^{(k)}$	2	5	14	41	122	365
	1	4	13	40	121	364
$\lambda^{(k)}$		2,5	2,8	2,93	2,98	2,99

Comme le montre l'exemple, il arrive que certaines composantes de  $\mathbf{u}^{(i)}$  deviennent très grandes (ou très petites) au cours du calcul. Pour éviter tout problème de dépassement de capacité, on peut normaliser  $\mathbf{x}^{(i)}$  de telle façon que  $\sup_m |x_m^{(i)}| = 1$ , avant de calculer  $\mathbf{x}^{(i+1)}$  (normalisation au sens de la norme infinie).

## 2.2 Puissance n-ième avec décalage

La rapidité de convergence de l'algorithme précédent dépend des rapports  $\lambda_i/\lambda_1$ . La convergence sera lente s'il existe une autre valeur propre proche de  $\lambda_1$ . Il est possible d'accélérer la convergence, en augmentant les différences entre valeurs propres. Il suffit de remarquer que  $\mathbf{A}$  et  $\mathbf{A}' = \mathbf{A} - s\mathbf{I}$  ont mêmes vecteurs propres (ces matrices commutent) et que leurs valeurs propres diffèrent par une translation. Si  $\mathbf{u}$  et  $\lambda$  sont éléments propres de  $\mathbf{A}$ , alors :

$$(\mathbf{A} - s\mathbf{I})\mathbf{u} = \mathbf{A}\mathbf{u} - s\mathbf{I}\mathbf{u} = (\lambda - s)\mathbf{u}$$

Ce procédé est utile dans le cas  $\lambda_1 \cong -\lambda_2$  ; après changement d'origine,  $\lambda'_1$  pourra être très différent de  $\lambda'_2$ .

## 2.3 Puissance n-ième de l'inverse

Je peux, en principe, obtenir une estimation de la plus petite valeur propre de  $\mathbf{A}$ , en module, en multipliant  $n$  fois le vecteur  $\mathbf{x}^{(0)}$  par la matrice  $\mathbf{A}^{-1}$ . Je vous laisse les lecteurs démontrer ce résultat. Dans la pratique, on ne calcule pas  $\mathbf{x}^{(n)} = \mathbf{A}^{-1}\mathbf{x}^{(n-1)}$ , mais on résout le système linéaire  $\mathbf{A}\mathbf{x}^{(n)} = \mathbf{x}^{(n-1)}$ , ce que l'on peut faire rapidement en formant une fois pour toute la décomposition  $\mathbf{LU}$  de  $\mathbf{A}$ . Cette méthode a pour mérite principal de servir d'introduction à l'algorithme suivant, plus utile.

## 2.4 Puissance n-ième de l'inverse avec décalage

Considérons la matrice  $\mathbf{B} = (\mathbf{A} - s\mathbf{I})^{-1}$  ; quels sont ses éléments propres ? Si  $s$  ne coïncide pas avec l'une des valeurs propres de  $\mathbf{A}$ ,  $\mathbf{B}$  est régulière et commute avec  $\mathbf{A}$ . Elle admet donc les mêmes vecteurs propres. Soit  $\mathbf{u}$  l'un de ces vecteurs ; il obéit à la relation  $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$ , ou encore  $\mathbf{u} = \lambda\mathbf{A}^{-1}\mathbf{u}$ , ce qui montre que  $1/\lambda$  est valeur propre de  $\mathbf{A}^{-1}$  ; en combinant ce résultat avec les propriétés de  $\mathbf{A} - s\mathbf{I}$ , je vois que  $1/(\lambda - s)$  est valeur propre de  $\mathbf{B}$ . Étant donné un vecteur initial  $\mathbf{x}^{(0)}$ , je forme par récurrence :

$$\mathbf{x}^{(k)} = \mathbf{B}\mathbf{x}^{(k-1)} = \mathbf{B}^k\mathbf{x}^{(0)} = (\mathbf{A} - s\mathbf{I})^{-k}\mathbf{x}^{(0)}.$$

Développons  $\mathbf{x}^{(0)}$  selon les vecteurs propres de  $\mathbf{A}$ , en appelant  $\lambda_m$  la valeur propre la plus proche de  $s$  :

$$\mathbf{x}^{(k)} = \sum_i \frac{c_i}{(\lambda_i - s)^k} \mathbf{u}_i = \frac{c_m}{(\lambda_m - s)^k} \left[ \mathbf{u}_m + \sum_{i \neq m} c_i \left( \frac{\lambda_m - s}{\lambda_i - s} \right)^k \mathbf{u}_i \right]$$

Comme  $|\lambda_m - s| < |\lambda_i - s|$ , le vecteur  $\mathbf{x}^{(k)}$  converge vers un vecteur proportionnel à  $\mathbf{u}^{(m)}$ . Ici encore, il ne faut pas calculer les inverses des matrices, mais résoudre des systèmes linéaires, comme  $(\mathbf{A} - s\mathbf{I})\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)}$ .

Cet algorithme peut me permettre de calculer toutes les valeurs propres de  $\mathbf{A}$ . Il suffit de balayer, à l'aide du paramètre  $s$ , l'intervalle où doivent se trouver les valeurs propres. Pour chaque valeur de  $s$ , l'itération sur  $\mathbf{x}$  converge vers le vecteur propre appartenant à la valeur propre la plus proche de  $s$  ; par ailleurs, le rapport de deux coordonnées permet le calcul de cette valeur propre :  $x_j^{(k)}/x_j^{(k-1)} = 1/(\lambda_j - s)$ .

En fait, la méthode s'avère peu pratique pour la recherche systématique des valeurs propres ; elle est au contraire très commode pour déterminer les vecteurs propres, connaissant des valeurs

approchées des  $\lambda_i$ . Supposons en effet qu'une autre méthode, différente, m'ait fourni un nombre  $\lambda'$ , valeur approchée au millième de  $\lambda_m$ . Si je choisis  $s = \lambda'$ , l'algorithme précédent va converger en quelques itérations vers le vecteur propre associé à  $\lambda_m$ , à condition que toutes les autres valeurs propres diffèrent de  $\lambda_m$  par plus de quelques millièmes.

*Exemple.* Au §2.1, j'ai trouvé une valeur approchée de la plus grande valeur propre d'une matrice  $\mathbf{A}$ , soit  $s = 2,99$ . Cherchons le vecteur propre correspondant à partir de  $\mathbf{x}^{(0)} = (1, 0)^T$ . Il me faut résoudre le système :

$$\begin{aligned} -1,01u + v &= 1, \\ u - 1,01v &= 0, \end{aligned}$$

dont la solution est  $[-50,2487; -49,7512]^T$  ou  $[1; 0,990099]^T$  après normalisation. Une autre itération donnerait le vecteur  $[1; 0,99995]^T$  exact à  $5 \cdot 10^{-5}$  près.

### 3 Méthode de Jacobi

Contrairement aux algorithmes précédents, la méthode de Jacobi permet le calcul de tous les éléments propres simultanément. Il s'agit d'effectuer une série de similitudes qui vont amener la matrice carrée symétrique  $\mathbf{A}$  progressivement à la forme diagonale, tout en préservant ses valeurs propres.

#### 3.1 Principe

Le principe de la méthode est facile à comprendre sur le cas à deux dimensions, et c'est par là que je commence. Soit la matrice :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$$

que je souhaite diagonaliser par une similitude  $\mathbf{A}' = \mathbf{R}^T \mathbf{A} \mathbf{R}$ , où  $\mathbf{R}$  est une matrice orthogonale qui s'écrit :

$$\mathbf{R} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Un calcul élémentaire montre que :

$$\mathbf{A}' = \begin{bmatrix} a_{11}c^2 - 2a_{12}sc + a_{22}s^2 & a_{12}(c^2 - s^2) + (a_{22} - a_{11})cs \\ a_{12}(c^2 - s^2) + (a_{22} - a_{11})cs & a_{11}s^2 + 2a_{12}cs + a_{22}c^2 \end{bmatrix}.$$

Je peux alors diagonaliser  $\mathbf{A}$  en choisissant  $\alpha$  selon la relation :

$$\operatorname{tg} 2\alpha = 2a_{12}/(a_{22} - a_{11}).$$

Ces petits calculs admettent une interprétation géométrique simple. Soit  $\mathbf{x} = (x_1, x_2)$  un vecteur de  $\mathbb{R}^2$  ; à la matrice  $\mathbf{A}$ , j'associe la forme quadratique  $\mathbf{x}^T \mathbf{A} \mathbf{x}$  et l'équation :

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = 1 = a_{11}x_1^2 + (a_{12} + a_{21})x_1x_2 + a_{22}x_2^2$$

qui représente une conique ; je suppose, pour fixer les idées, que cette courbe est une ellipse. J'effectue maintenant le changement de repère défini par :

$$\mathbf{x} = \mathbf{R}\mathbf{y}.$$

Il s'agit d'une rotation des axes d'angle  $\alpha$ . Je fais le changement de variables dans l'équation de l'ellipse :

$$(\mathbf{R}\mathbf{y})^T \mathbf{A}\mathbf{R}\mathbf{y} = \mathbf{y}\mathbf{R}^T \mathbf{A}\mathbf{R}\mathbf{y} = \mathbf{y}\mathbf{A}'\mathbf{y} = 1$$

La nouvelle équation de la conique est définie par la nouvelle matrice. Si  $\mathbf{A}'$  est diagonale parce que j'ai fait un bon choix pour  $\alpha$ , l'équation s'écrit :

$$a'_{11}y_1^2 + a'_{22}y_2^2 = 1.$$

On dit que la conique «a été ramenée à ses axes principaux» ; les axes principaux sont aussi les axes de symétrie de l'ellipse. Les termes «croisés» ou «rectangles» ont disparu. La généralisation de ce procédé à  $n$  dimensions constitue l'algorithme de Jacobi.

### 3.2 Méthode de Jacobi

Je vais appliquer à la matrice  $\mathbf{A}$  une suite de similitudes  $\mathbf{A} \rightarrow \mathbf{A}^{(1)} \rightarrow \mathbf{A}^{(2)} \rightarrow \dots \rightarrow \mathbf{A}^{(n)}$ , avec  $\mathbf{A}^{(i)} = \mathbf{R}_i^{-1} \mathbf{A}^{(i-1)} \mathbf{R}_i$  de telle manière que  $\mathbf{A}^{(n)}$  tende vers une matrice diagonale  $\mathbf{D}$  d'éléments diagonaux  $\lambda_i$ . Il est facile de comprendre pourquoi la diagonalisation n'est qu'approchée. A l'aide d'une rotation des axes du plan  $(x_1, x_2)$ , je peux annuler l'élément  $a_{12}$  comme au paragraphe précédent. Mais, lorsque que j'applique une nouvelle similitude pour faire disparaître  $a_{13}$ , tous les éléments de la ligne 1, en particulier le zéro que je viens de créer en  $a_{12}$ , sont modifiés. Ce n'est que progressivement que la "substance" de la matrice va se concentrer sur la diagonale.  $\mathbf{A}$  est symétrique, réelle. Le passage de  $\mathbf{A}^{(i)}$  (éléments  $a_{rs}$ ) à  $\mathbf{A}^{(i+1)}$  (d'éléments  $a'_{rs}$ ) fait intervenir la matrice orthogonale  $\mathbf{R}_{j,k}$  qui vaut (en posant  $c = \cos \theta_{jk}$  et  $s = \sin \theta_{jk}$ ) :

$$\mathbf{R}_{j,k} = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & & & & & \dots & 0 \\ 0 & & 1 & & & & & \vdots \\ \vdots & & & c & & s & & \vdots \\ \vdots & & & & 1 & & & \vdots \\ \vdots & & & -s & & c & & \vdots \\ 0 & & & & & & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 1 \end{bmatrix}$$

$\mathbf{R}_{j,k}$  représente une rotation des axes dans le sous-espace  $(j, k)$ . Seules les colonnes  $j$  et  $k$  et les lignes  $j$  et  $k$  de  $\mathbf{A}^{(i+1)}$  diffèrent de celles de  $\mathbf{A}^{(i)}$ . En calculant d'abord  $\mathbf{A}\mathbf{R}$ , puis  $\mathbf{R}^T \mathbf{A}\mathbf{R}$ , vous montrerez que les équations de transformation s'écrivent :

$$\begin{aligned} a'_{rs} &= a_{rs}, \quad r, s \neq j, k, \\ a'_{rj} &= a'_{jr} = ca_{rj} - sa_{rk}, \quad r \neq j, k, \\ a'_{rk} &= a'_{kr} = sa_{rj} + ca_{rk}, \quad r \neq j, k, \\ a'_{jj} &= c^2 a_{jj} + s^2 a_{kk} - 2csa_{jk}, \\ a'_{jk} &= a'_{kj} = cs(a_{jj} - a_{kk}) + (c^2 - s^2)a_{jk}, \\ a'_{kk} &= s^2 a_{jj} + c^2 a_{kk} + 2csa_{jk}. \end{aligned}$$

L'angle de rotation est défini par la condition  $a'_{jk} = 0$

$$\operatorname{tg} 2\theta_{jk} = (a_{kk} - a_{jj}) / (2a_{jk}).$$

En pratique, on effectue les calculs sur une moitié de la matrice, puisque la similitude respecte la symétrie. On ne calcule pas non plus  $a'_{jk}$ , dont on sait qu'il est nul.

Je peux donner une indication sur la convergence de la méthode. Nous définissons les normes extradiagonales de  $\mathbf{A}$  et de  $\mathbf{A}'$  par les relations :

$$S^2 = \sum_{j \neq k} |a_{jk}|^2; S'^2 = \sum_{j \neq k} |a'_{jk}|^2.$$

En remplaçant les  $a'_{jk}$  par leurs expressions, on montre que  $S' < S$ , si bien que la suite des  $S(\mathbf{A}^{(j)})$ , positive décroissante, converge. Il est un peu plus difficile de montrer que les  $S(\mathbf{A}^{(j)})$  tendent vers zéro, à condition que les éléments à éliminer soient choisis dans l'ordre "lexicographique"  $a_{12}, a_{13}, \dots, a_{1n}, a_{23}, \dots, a_{2n}, \dots$ . On arrête le calcul quand  $\sup |a_{ij}| < \text{seuil}$ .

Lorsque la convergence est atteinte, la matrice est réputée diagonale et  $\mathbf{Q}^T \mathbf{A} \mathbf{Q} = \mathbf{D}$ , avec  $\mathbf{R}_1 \mathbf{R}_2 \dots \mathbf{R}_n = \mathbf{Q}$ . Les colonnes de  $\mathbf{Q}$  sont les vecteurs propres et nous intéressent. Il serait ruineux de conserver les matrices  $\mathbf{R}_i$  pour les multiplier à la fin. Je procède en fait par récurrence, en posant

$$\mathbf{Q}_k = \mathbf{R}_1 \mathbf{R}_2 \dots \mathbf{R}_k, k < n$$

et

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k \mathbf{R}_{k+1}$$

. Chaque fois que je définis les éléments d'une matrice  $\mathbf{R}$ , j'en profite pour former la matrice  $\mathbf{Q}_{k+1}$  (éléments  $q'_{rs}$ ) à partir de  $\mathbf{Q}_k$  (éléments  $q_{rs}$ ) :

$$\begin{aligned} q'_{rs} &= q_{rs}, & r, s \neq j, k, \\ q'_{rj} &= cq_{rj} + sq_{rk}, & r \neq j, k, \\ q'_{rk} &= -sq_{rj} + cq_{rk}, & r \neq j, k; \end{aligned}$$

Dès que l'ordre de la matrice dépasse une dizaine, la méthode de Jacobi est moins rapide que la réduction de la matrice à la forme tridiagonale, suivie d'un calcul de valeurs propres. Cette remarque est d'autant plus pertinente que  $\mathbf{A}$  a la forme d'une matrice bande plus ramassée : la méthode de Jacobi détruit cette structure.

*Exemple.*

Je me propose de diagonaliser la matrice symétrique

$$\mathbf{A} = \mathbf{A}^0 = \begin{bmatrix} 1.0 & 1. & 0.5 \\ 1.0 & 1. & 0.25 \\ 0.5 & 0.25 & 2.0 \end{bmatrix}$$

Je calcule d'abord le carré de la norme diagonale  $\rightarrow \text{sum}(\mathbf{A} \cdot \mathbf{A}) = 8.625$  et le carré de la norme complète de  $\mathbf{A}$  :  $\rightarrow \text{trace}(\mathbf{A} \cdot \mathbf{A}) = 6.0$  Pour annuler l'élément  $a_{12}$ , il faut effectuer une première rotation dans le sous-espace (1,2). Comme  $a_{22} = a_{11}$ , j'ai  $2\theta = \pi/2, \theta = \pi/4$ , si bien que

$$\mathbf{R}_1 = \begin{bmatrix} .7071068 & .7071068 & 0. \\ -.7071068 & .7071068 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

et

$$\mathbf{A}^1 = \mathbf{R}_1^T \mathbf{A} \mathbf{R}_1 = \begin{bmatrix} 0. & 0. & .1767767 \\ 0. & 2. & .5303301 \\ .1767767 & .5303301 & 2. \end{bmatrix}$$

La norme au carré reste constante,  $\text{sum}(\mathbf{A}1.*\mathbf{A}1) = 8.625$ , alors que la somme des carrés des éléments diagonaux augmente :  $\text{trace}(\mathbf{A}1.*\mathbf{A}1) = 8$ . La deuxième rotation, destinée à annuler le plus grand élément extradiagonal, soit  $a_{23}^1$ , est encore particulière,  $\theta = \pi/4$  et

$$\mathbf{R}_2 = \begin{bmatrix} 1. & 0. & 0. \\ 0. & .7071068 & .7071068 \\ 0. & -.7071068 & .7071068 \end{bmatrix}$$

$$\mathbf{A}^2 = \mathbf{R}_2^T * \mathbf{A}^1 * \mathbf{R}_2 = \begin{bmatrix} 0. & -.125 & .125 \\ -.125 & 1.4696699 & 2.220E-16 \\ .125 & 2.220E-16 & 2.5303301 \end{bmatrix}$$

Les tous petits nombres  $a_{23}^2 = a_{32}^2$  sont dûs à des erreurs d'arrondi ; je fait le ménage pour les faire disparaître

$$\mathbf{A}2 = \text{clean}(\mathbf{A}2) = \begin{array}{ccccc} ! & 0. & - & .125 & .125 & ! \\ & ! & - & .125 & 1.4696699 & 0. & ! \\ & & ! & .125 & 0. & 2.5303301 & ! \end{array}$$

Je surveille le carré de la norme complète,  $\text{sum}(\mathbf{A}2.*\mathbf{A}2) = 8.625$  et le carré de la norme diagonale,  $\text{trace}(\mathbf{A}2.*\mathbf{A}2) = 8.5625$ . Vous remarquez que les éléments précédemment annulés ont tendance à «repousser», bien que moins vigoureusement.

La troisième rotation, dans le sous-espace (1,2), vise à éliminer  $a_{12}^2$ . Je trouve  $\text{tg } 2\theta = 2a_{12}^2/(a_{22}^2 - a_{11}^2) = -0.1701062$ , d'où la matrice de rotation

$$\mathbf{R}_3 = \begin{bmatrix} .9964533 & -.0841471 & 0. \\ .0841471 & .9964533 & 0. \\ 0. & 0. & 1. \end{bmatrix}$$

et la matrice  $\mathbf{A}^3$ , débarrassée d'éléments inférieurs à  $10^{-16}$  :

$$\mathbf{A}^3 = \begin{bmatrix} -.0105558 & 0. & .1245567 \\ 0. & 1.4802257 & -.0105184 \\ .1245567 & -.0105184 & 2.5303301 \end{bmatrix}$$

Les éléments diagonaux se renforcent :  $\text{trace}(\mathbf{A}3.*\mathbf{A}3) = 8.59375$ .

La quatrième rotation aura lieu dans le sous-espace (1,3), avec  $\text{tg } 2\theta = 0.0980419$  ou  $\theta = 0.0488648$  rd. La matrice de rotation s'écrit

$$\mathbf{R}_4 = \begin{bmatrix} .9988064 & 0. & .0488453 \\ 0. & 1. & 0. \\ -.0488453 & 0. & .9988064 \end{bmatrix}$$

et permet d'obtenir

$$\mathbf{A}^4 = \begin{bmatrix} -.0166471 & .0005138 & 0. \\ .0005138 & 1.4802257 & -.0105058 \\ 0. & -.0105058 & 2.5364214 \end{bmatrix}$$

Mon critère de convergence est atteint : j'arrête l'itération. Le carré de la norme diagonale vaut maintenant 8.6247788 (à  $3 \cdot 10^{-4}$  du total). Les vecteurs propres sont les colonnes de la matrice de similitude globale ; comme je n'ai à manipuler que 4 petites matrices, je forme  $\mathbf{R}$  directement

$$\mathbf{R} = \mathbf{R}_1 * \mathbf{R}_2 * \mathbf{R}_3 * \mathbf{R}_4 = \begin{bmatrix} .7213585 & .4387257 & .5358747 \\ -.6861572 & .5577276 & .4670419 \\ -.0939688 & -.7045989 & .7033564 \end{bmatrix}$$

Cette matrice est orthogonale aux erreurs d'arrondi près, comme vous le vérifierez en formant  $\mathbf{R}^T * \mathbf{R}$ . Je vous recommande aussi d'effectuer le produit de  $\mathbf{A}$  par une colonne de  $\mathbf{R}$ , pour vérifier que l'on obtient bien un vecteur proportionnel à cette colonne et à la valeur propre correspondante.

Enfin, je peux demander à Scilab de faire les calculs directement, avec une tolérance de l'ordre de  $10^{-16}$  :

```
-->[D,X] = bdiag(A)
X =
!      .7212071  - .5314834  - .4442811  !
! - .6863493  - .4614734  - .5621094  !
! - .0937280  - .7103293  .6976011  !
D =
! - .0166473  0.          0.          !
!  0.         2.5365259  0.          !
!  0.         0.         1.4801214  !
```

L'erreur encourue sur les valeurs propres après 4 itérations de l'algorithme de Jacobi n'excède pas  $10^{-4}$ . Les vecteurs propres sont nettement moins précis. C'est toujours le cas pour l'algorithme de Jacobi. Une dernière remarque : l'ordre dans lequel apparaissent les valeurs propres et les vecteurs propres associés est arbitraire, de même que le signe des vecteurs propres.

## 4 Réduction à la forme tridiagonale

Il est avantageux d'abandonner l'objectif de l'algorithme de Jacobi (diagonaliser une matrice symétrique par une suite convergente de similitudes) pour un objectif plus restreint : amener la matrice réelle symétrique  $\mathbf{A}$  à la forme tridiagonale. En effet, la détermination des éléments propres d'une matrice tridiagonale est très rapide. D'autre part, la transformation de  $\mathbf{A}$  en matrice tridiagonale nécessite, grâce à l'algorithme de Householder, un nombre fini d'opérations.

### 4.1 Matrice de Householder

L'algorithme de Householder se comprend facilement à partir de sa représentation géométrique dans l'espace à trois dimensions. Soient un point  $M$  et un plan  $(P)$ , contenant l'origine (mais pas  $M$ ) et normal au vecteur unitaire  $\hat{\mathbf{n}}$ . Je cherche à construire le point  $M'$  qui se déduit de  $M$  par une symétrie orthogonale par rapport à  $(P)$ . La droite passant par  $M$  et perpendiculaire à  $(P)$  perce ce plan en  $K$  ; elle passe aussi par  $M'$ . Le vecteur  $\overrightarrow{KM}$  est parallèle à  $\hat{\mathbf{n}}$  ; sa longueur est celle de la projection orthogonale de  $OM$  sur  $\hat{\mathbf{n}}$  ; autrement dit,  $\overrightarrow{KM} = (\hat{\mathbf{n}} \cdot \overrightarrow{OM})\hat{\mathbf{n}}$ . Par symétrie,  $\overrightarrow{KM'}$  est l'opposé de  $\overrightarrow{KM}$ . Je peux relier  $M'$  à  $M$  par  $\overrightarrow{OM'} = \overrightarrow{OM} + \overrightarrow{MM'} = \overrightarrow{OM} - 2\overrightarrow{KM} = \overrightarrow{OM} - 2(\hat{\mathbf{n}} \cdot \overrightarrow{OM})\hat{\mathbf{n}}$ . Il est clair que  $OM' = OM$ , la symétrie préservant les longueurs.



Lorsque je représente les vecteurs par des matrices colonne, les relations précédentes deviennent  $\overrightarrow{OM} = \mathbf{x}$ ,  $\overrightarrow{OM'} = \mathbf{x}'$  et  $\mathbf{x}' = \mathbf{x} - 2(\hat{\mathbf{n}}^T \mathbf{x})\hat{\mathbf{n}} = \mathbf{x} - 2\hat{\mathbf{n}}\hat{\mathbf{n}}^T \mathbf{x} = (\mathbf{I} - 2\hat{\mathbf{n}}\hat{\mathbf{n}}^T)\mathbf{x}$ , les dernières formes étant rendues possibles par l'associativité des produits.

Ce raisonnement se généralise sans peine à l'espace à  $n$  dimensions, même si la représentation géométrique devient plus difficile. Soit  $\mathbf{w}$  un vecteur unitaire de  $\mathbb{R}^n$  :  $\mathbf{w}^T \mathbf{w} = 1$ . Je définis une matrice :

$$\mathbf{P} = \mathbf{I} - 2\mathbf{w}\mathbf{w}^T.$$

Cette matrice est symétrique :  $\mathbf{P}^T = \mathbf{P}$  parce que le transposé d'un produit est égal au produit des transposées dans l'ordre inverse. Elle est aussi orthogonale :

$$\begin{aligned} \mathbf{P}^T \mathbf{P} &= \mathbf{P} \mathbf{P} = (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)(\mathbf{I} - 2\mathbf{w}\mathbf{w}^T) \\ &= \mathbf{I} - 2\mathbf{w}\mathbf{w}^T - 2\mathbf{w}\mathbf{w}^T + 4\mathbf{w}\mathbf{w}^T \mathbf{w}\mathbf{w}^T = \mathbf{I}. \end{aligned}$$

et donc idempotente  $\mathbf{P}^2 = \mathbf{I}$ . Si deux vecteurs  $\mathbf{x}, \mathbf{y}$  satisfont à la relation :

$$\mathbf{y} = \mathbf{P}\mathbf{x} = \mathbf{x} - 2(\mathbf{w}\mathbf{w}^T \mathbf{x}),$$

alors :

$$\mathbf{y}^T \mathbf{y} = \mathbf{x}^T \mathbf{P}^T \mathbf{P} \mathbf{x} = \mathbf{x}^T \mathbf{x}.$$

Je m'intéresse maintenant à la réciproque du cas précédent : je voudrais déterminer un vecteur  $\mathbf{w}$  (et donc une matrice  $\mathbf{P}$ ) tel qu'un  $\mathbf{x}$  donné soit transformé par  $\mathbf{P}$  en un vecteur donné à l'avance et même, plus précisément, en un multiple du premier vecteur de la base canonique

$$\mathbf{P}\mathbf{x} = k\mathbf{e}_1.$$

D'après les relations précédentes, il apparaît que  $|k|^2 = \|\mathbf{x}\|^2 = \mathbf{x}^T \mathbf{x} \equiv L^2$  et  $k = \pm L$ . De plus, d'après la définition de  $\mathbf{P}\mathbf{x}$ ,  $\mathbf{x} - k\mathbf{e}_1 = 2\mathbf{w}\mathbf{w}^T \mathbf{x} = 2(\mathbf{w}^T \mathbf{x})\mathbf{w}$  : le vecteur  $\mathbf{w}$  doit être proportionnel à  $\mathbf{x} - k\mathbf{e}_1$ . Comme il doit être en plus normalisé, je peux écrire

$$\mathbf{w} = \frac{\mathbf{x} - k\mathbf{e}_1}{\|\mathbf{x} - k\mathbf{e}_1\|}$$

De plus

$$\|\mathbf{x} - k\mathbf{e}_1\| = \|\mathbf{x} \pm L\mathbf{e}_1\| = [(x_1 \pm L)^2 + x_2^2 + \dots + x_n^2]^{1/2}.$$

On montre qu'il vaut mieux choisir le signe moins (pour éviter des erreurs d'arrondi), si bien que :

$$|x_1 - k|^2 = |x_1 + L|^2 = L^2 + 2L|x_1| + x_1^2.$$

Il s'en suit que :  $\|\mathbf{x} - k\mathbf{e}_1\|^2 = 2L^2 + 2L|x_1|$ . On peut écrire, sans normaliser le vecteur  $\mathbf{w}$  :

$$\mathbf{P} = \mathbf{I} - \beta \mathbf{u}\mathbf{u}^T$$

avec :  $L = \|\mathbf{x}\|$ ,  $k = -L$ ,  $\mathbf{u} = \mathbf{x} - k\mathbf{e}_1 = [x_1 + L, x_2, \dots, x_n]^T$ ,  $\beta = 1/L(L + x_1)$ .

*Exemple.* Soit le vecteur  $\mathbf{x} = [1, -2, 3, 1, -1]^T$  dont la norme est  $L = 4$ . Le vecteur  $\mathbf{u}$  vaut  $[5, -2, 3, 1, -1]^T$  et  $\beta = 1/20$ . On trouve alors :

$$\mathbf{P} = \frac{1}{20} \begin{bmatrix} -5 & 10 & -15 & -5 & 5 \\ 10 & 16 & 6 & 2 & -2 \\ -15 & 6 & 11 & -3 & 3 \\ -5 & 2 & -3 & 19 & 1 \\ 5 & -2 & 3 & 1 & 19 \end{bmatrix}$$

et on vérifie que  $\mathbf{P}\mathbf{x} = [-4, 0, 0, 0, 0]^T$ .

## 4.2 Tridiagonalisation

Je vais utiliser des matrices de Householder comme  $\mathbf{P}$  pour amener la matrice réelle symétrique  $\mathbf{A}$  à la forme tridiagonale, à l'aide d'une série de similitudes :

$$\mathbf{A}^{(i)} = \mathbf{Q}^{(i)T} \mathbf{A}^{(i-1)} \mathbf{Q}^i.$$

Le premier produit (prémultiplication qui agit sur les colonnes),  $\mathbf{Q}^T \mathbf{A}$ , doit laisser invariant  $a_{11}$  et doit transformer le vecteur  $[a_{21}, a_{31}, \dots, a_{n1}]^T$  en  $[k, 0, 0, \dots]^T$ , un multiple du premier vecteur de la base canonique de  $\mathbb{R}^{n-1}$ . Il faut donc faire agir une matrice de Householder  $\mathbf{P}_{n-1}$  de dimension  $n-1$  ; celle-ci sera insérée dans une matrice d'ordre  $n$  ayant la structure

$$\mathbf{Q} = \left[ \begin{array}{c|ccc} 1 & 0 & \dots & 0 \\ \hline 0 & \mathbf{P}_{n-1} & & \end{array} \right]$$

Le deuxième produit  $(\mathbf{Q}^T \mathbf{A}) \mathbf{Q}$  (postmultiplication agissant sur les lignes) fera de même pour la première ligne de  $\mathbf{A}$  :  $a_{11}$  invariant et  $[a_{12}, a_{13}, \dots, a_{1n}]$  transformé en  $[k, 0, \dots, 0]$ .

Vous imaginez sans peine la suite de l'itération : à l'aide d'une matrice de Householder  $n-2 \times n-2$ , je forme une nouvelle matrice  $\mathbf{Q}$ , d'ordre  $n-1$ , dont la mission sera d'annuler les éléments  $a_{42}, \dots, a_{n2}$  et  $a_{24}, \dots, a_{2n}$  (ce sont les mêmes). Au bout de  $n-2$  similitudes, la matrice sera exactement sous forme tridiagonale : cet algorithme est plus compliqué que celui de Jacobi, mais il aboutit au résultat cherché en un nombre fini d'opérations.

Il faudra, en principe, garder les matrices  $\mathbf{Q}^i$  pour le calcul des vecteurs propres. Si  $\mathbf{y}$  est un vecteur propre de  $\mathbf{A}_{n-2}$ ,  $\mathbf{x} = \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_{n-2} \mathbf{y}$  est vecteur propre de  $\mathbf{A}$ . En pratique, le produit des  $\mathbf{Q}$  est accumulé au fur et à mesure.

*Exemple.* Je vais tridiagonaliser la matrice qui m'a déjà servi à illustrer la méthode de Jacobi :

$$\mathbf{A} = \begin{bmatrix} 1. & 1. & .5 \\ 1. & 1. & .25 \\ .5 & .25 & 2. \end{bmatrix}$$

Le bas de la première colonne constitue le vecteur à traiter

$$\mathbf{x} = A(2:3, 1) = \begin{bmatrix} 1. \\ .5 \end{bmatrix}.$$

Je calcule successivement  $k = \text{sqrt}(\mathbf{x}' * \mathbf{x}) = 1.118034$ ,

$$\mathbf{w} = \mathbf{x} - k * [10]' = \begin{bmatrix} -.1180340 \\ .5 \end{bmatrix},$$

que je normalise

$$\mathbf{w} = \mathbf{w} / \text{sqrt}(\mathbf{w}' * \mathbf{w}) = \begin{bmatrix} -.2297529 \\ .9732490 \end{bmatrix}.$$

J'en déduit

$$\mathbf{P} = \text{eye}(2, 2) - 2 * \mathbf{w} * \mathbf{w}' = \begin{bmatrix} .8944272 & .4472136 \\ .4472136 & -.8944272 \end{bmatrix}.$$

Je vérifie ce premier résultat

$$\mathbf{P} * A(2:3, 1) = \mathbf{P} \mathbf{x} = \begin{bmatrix} 1.118034 \\ 2.776E-16 \end{bmatrix}.$$

Je construit la matrice  $\mathbf{Q}$  avant de tridiagonaliser :

$$\mathbf{Q} = [1 \ 0 \ 0; 0 \ P(1,1) \ P(1,2); 0 \ P(2,1) \ P(2,2)] = \begin{bmatrix} 1. & 0. & 0. \\ 0. & .8944272 & .4472136 \\ 0. & .4472136 & -.8944272 \end{bmatrix}.$$

Pour une matrice  $3 \times 3$ , il n'y a qu'une étape de tridiagonalisation :

$$\mathbf{A1} = \mathit{clean}(\mathbf{A1}) = \begin{bmatrix} 1. & 1.118034 & 0. \\ 1.118034 & 1.4 & -.55 \\ 0. & -.55 & 1.6 \end{bmatrix}.$$

Quelques vérifications élémentaires :  $\mathit{trace}(\mathbf{A1}) = 4. = \mathit{trace}(\mathbf{A})$  et  $\mathit{det}(\mathbf{A1}) = -.0625 = \mathit{det}(\mathbf{A})$ .

### 4.3 Calcul des valeurs propres

Soit  $\mathbf{J}$  une matrice tridiagonale symétrique réelle d'ordre  $n$  :

$$\mathbf{J} = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \delta_2 & \gamma_3 & \cdots & 0 \\ 0 & \gamma_3 & \delta_4 & \gamma_4 & \cdots \\ \vdots & & \ddots & \ddots & \\ & & & \gamma_n & \delta_n \end{bmatrix}.$$

Nous supposons les  $\gamma_i$  tous différents de zéro. Nous pouvons former facilement le polynôme caractéristique de  $\mathbf{J}$ , en procédant par récurrence. Définissons en effet la matrice partielle :

$$\mathbf{J}_i = \begin{bmatrix} \delta_1 & \gamma_2 & 0 & \cdots \\ \gamma_2 & \delta_1 & \gamma_3 & \cdots \\ \cdots & & & \cdots \\ & & & \gamma_i \\ & & & \gamma_i & \delta_i \end{bmatrix}$$

et

$$p_i(x) = \mathit{det}[\mathbf{J}_i - \lambda \mathbf{I}].$$

En développant ce déterminant en fonction des éléments de la dernière colonne, nous trouvons les relations :

$$p_0(x) = 1; p_1(x) = \delta_1 - x;$$

$$p_i(x) = (\delta_i - x)p_{i-1}(x) - \gamma_i^2 p_{i-2}(x).$$

Toute méthode efficace de recherche des zéros d'un polynôme est maintenant applicable au calcul des valeurs propres (les racines de  $p_n$ ) ; la méthode de bisection est stable et commode. On peut aussi utiliser l'algorithme de Newton, en calculant la dérivée de  $p_n$  à l'aide de la récurrence :

$$p'_0(x) = 0; p'_1(x) = -1,$$

$$p'_i(x) = -p_{i-1}(x) + (\delta_i - x)p'_{i-1}(x) - \delta_i^2 p'_{i-2}(x).$$

## 5 Matrices hermitiques

On rencontre souvent, en physique, des matrices hermitiques : comment chercher leurs éléments propres ? La méthode de Householder peut s'étendre à des matrices complexes, mais la programmation en Pascal est pénible, puisque ce langage ignore le TYPE complexe et que l'on est obligé de réécrire toutes les opérations arithmétiques (ce ne serait pas le cas en FORTRAN ou en C). Une autre méthode consiste à séparer partie réelle et partie imaginaire du problème de valeurs propres. Je cherche des solutions de :

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x},$$

où  $\mathbf{A}$  est hermitique et  $\lambda$  est réel. Je pose  $\mathbf{A} = \mathbf{A}' + i\mathbf{A}''$ ,  $\mathbf{x} = \mathbf{x}' + i\mathbf{x}''$ . L'hermiticité de  $\mathbf{A}$  impose la symétrie de  $\mathbf{A}'$  et l'antisymétrie de  $\mathbf{A}''$  ( $\mathbf{A}''^T = -\mathbf{A}''$ ). En séparant partie réelle et partie imaginaire :

$$\begin{aligned} \mathbf{A}'\mathbf{x}' - \mathbf{A}''\mathbf{x}'' &= \lambda\mathbf{x}', \\ \mathbf{A}''\mathbf{x}' + \mathbf{A}'\mathbf{x}'' &= \lambda\mathbf{x}''. \end{aligned}$$

Ces relations peuvent être considérées comme le développement par blocs de l'équation :

$$\begin{bmatrix} \mathbf{A}' & -\mathbf{A}'' \\ \mathbf{A}'' & \mathbf{A}' \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \end{bmatrix}.$$

J'ai ainsi remplacé un problème  $n \times n$  complexe en un problème  $2n \times 2n$  réel. Le nombre de valeurs propres a-t-il doublé pour autant ?