Université de Bordeaux **Algorithmique mathématique 2**
Licence math-info Année 2017–2018

FEUILLE D'EXERCICES n° 11

Travail sur machine noté

Orthogonalisation de Gram-Schmidt

Il s'agit d'un travail exclusivement expérimental. Aucune justification théorique n'est demandée.

Dans l'énoncé, les indices de matrices et de vecteurs commencent à 0.

Soient E un espace euclidien. On note $(.|.)$ son produit scalaire et $||x|| = \sqrt{(x|x)}$. Soit $b = (b_0, \dots, b_{n-1})$ une base de E.

Il n'y a pas unicité de l'orthogonalisée de Gram-Schmidt, mais on peut imposer une condition pour obtenir cette unicité (en fait, il y a unicité pour chaque vecteur à multiplication près par un réel strictement positif). Le cours évoque les deux possibilités suivante pour cette condition supplémentaire (mais bien sur, il y a d'autres choix possibles).

(1) les coefficients diagonaux de la matrice de passage sont égaux à 1.

(2) l'orthogonalisée de Gram-Schmidt est orthonormale. On dira alors que c'est l'orthonormalisée de Gram-Schmidt associée à b.

Nous décrivons ci-dessous un algorithme pour chacune de ces possibilités.

Orthogonalisée de Gram-Schmidt de type (1) (algorithme vu en cours). Soit $b^* = (b_0^*, \dots, b_{n-1}^*)$ l'orthogonalisée de gram-schmidt de b telle que les coefficients diagonaux de la matrice de passage de b à b^* sont égaux à 1. Pour tout j, et pour tout $i \in [[0, j-1]]$, on pose

$$y_{ij} = \frac{(b_j|b_i^*)}{(b_i^*|b_i^*)}.$$

Alors

$$b_j^* = b_j - \sum_{i=0}^{j-1} y_{ij} b_i^*.$$

Si l'on pose $y_{ij} = 0$ pour $i > j$ et $y_{jj} = 1$ pour tout j, la matrice $U = (y_{ij})$ est la matrice de passage de b^* à b.

Orthonormalisée de Gram-Schmidt : type (2). Soit $b' = (b'_0, \dots, b'_{n-1})$ l'orthonormalisée de Gram-Schmidt de b. Pour calculer les b'_i, on pourrait calculer les b_i^*, puis $b'_i = \dfrac{b_i^*}{||b_i^*||}$ mais on va faire autrement : on va calculer ces b'_i directement en suivant la même méthode que pour b^*, mais en normalisant à chaque étape j le vecteur trouvé.

D'abord,

$$b'_0 = \frac{b_0}{||b_0||} \quad \text{et} \quad r_{00} = ||b_0||.$$

On suppose que l'on a construit $b'_0, \dots, b'_{j-1}$. Pour tout $i \in [[0, j-1]]$, on pose

$$r_{ij} = (b_j | b'_i)$$

puis

$$b''_j = b_j - \sum_{i=0}^{j-1} r_{ij} b'_i \quad \text{et} \quad r_{jj} = ||b''_j||$$

Ce vecteur b''_j n'est pas de norme 1 ; le vecteur b'_i que l'on veut calculer est

$$b'_i = \frac{b''_j}{r_{jj}}.$$

Enfin, si l'on pose $r_{ij} = 0$ pour $i > j$, la matrice $R = (r_{ij})$ est la matrice de passage de b' à b.

Exercice 1 – [ORTHOGONALISATION DE TYPE (1)]
Soit b une base de E et soit S une matrice symétrique définie positive. Soit $(.|.)$ le produit scalaire dont la matrice de Gram pour b est égal à S (alors si x et y sont deux éléments de E et si X et Y sont leurs coordonnées dans la base b, $(x|y) = {}^tXSY$).

Programmer une fonction `GS` qui prend en entrées b et S et qui en sortie rend l'orthogonalisée de Gram-Schmidt b^* de type (1) de b pour $(.|.)$, ainsi que la matrice de passage U de b^* à b.

Pour les premiers essais, on peut utiliser $S = I_2$ et une base b simple de $\mathbb{R}^2$.

Exercice 2 – [GRAM, SCHMIDT ET CHOLESKY]

1) Soit $S = \begin{pmatrix} 4 & -1 \\ -1 & 4 \end{pmatrix}$. En utilisant `GS`, calculer l'orthogonalisée e^* de type (1) de la base canonique e de $\mathbb{R}^2$ pour le produit scalaire de $\mathbb{R}^2$ dont la matrice de Gram dans e est S. Vérifier que la matrice de Gram de e^* est $D = \mathrm{Diag}(4, 15/4)$. Soit U la matrice de passage de e^* à e. Vérifier que $S = {}^tUDU$.

2) Écrire une fonction `Cho` qui étant donnée une matrice symétrique définie positive S rend la matrice U triangulaire supérieure aux coefficients diagonaux égaux à 1 et la matrice diagonale D telles que $S = {}^tUDU$.

3) Appliquer cette fonction à $S = (s_{ij}) \in M_{10}(\mathbb{R})$ où pour tout $(i, j) \in [[0, 9]]^2$,

$$s_{ij} = \min(i, j) + 1.$$

Exercice 3 – [UN EXEMPLE DE POLYNÔMES ORTHOGONAUX]

1) Soit $E = \mathbb{R}[x]_n$ l'ensemble des polynômes de degré inférieur ou égal à n, muni du produit scalaire

$$(P|Q) = \int_0^1 P(t)Q(t)dt.$$

On reconnaît l'exercice 6 de la feuille 9 (ou l'on a remplacé 2 par n). Le coefficient (i,j) de la matrice de Gram S de la base $b = (1, x, \ldots, x^n)$ de $\mathbb{R}[x]_n$ est

$$s_{ij} = \int_0^1 t^{i+j} = \frac{1}{i+j+1}.$$

Programmer une fonction `PolyOrtho` qui en entrée prend n et en sortie rend l'orthogonalisée de Gram-Schmidt de type (1) de b.
Note. Les polynômes obtenus sont dans $\mathbb{Q}[x]$. Pour définir cet anneau, on peut utiliser la commande

```
Qx.<x>=PolynomialRing(QQ)
```

Si v est le vecteur des coordonnées de P, on peut écrire P sous forme polynomiale par la commande

```
P=Qx(list(v))
```

2) Appliquer la fonction à $n = 5$. Soit $(P_0, \ldots, P_5)$ la base obtenue. Vérifier que pour tout $k \leqslant 5$,

$$P_k = \frac{k!}{(2k)!} \frac{d^k}{dx^k} \left((x(x-1))^k\right)$$

(voir la commande `diff` pour les dérivées k-èmes).
Remarque. En considérant d'autres produits scalaires du type

$$\int_a^b P(t)Q(t)\mu(t)dt$$

où μ est une fonction continue positive non identiquement nulle sur $]a, b[$, on trouve d'autres familles de polynômes orthogonaux (polynômes de Legendre, de Hermite, de Tchebycheff ...).

Exercice 4 – [Décomposition QR]
Soit $A \in \mathrm{GL}_n(\mathbb{R})$. Soient $A_0, \ldots, A_{n-1}$ les colonnes de A. On considère le produit scalaire $(X|Y) = {}^tXY$ dans $M_{n,1}(\mathbb{R})$. On note $(Q_0, \ldots, Q_{n-1})$ l'orthonormalisée de Gram-Schmidt de $(A_0, \ldots, A_{n-1})$. Soient $Q = (Q_0| \ldots |Q_{n-1})$ la matrice dont les vecteurs colonnes sont les Q_j et $R = (r_{ij}) \in M_n(\mathbb{R})$ où les $r_{i,j}$ sont définis dans l'algorithme du type (2) ci-dessus. Alors $Q \in \mathcal{O}_n(\mathbb{R})$, $R \in \mathcal{T}^{++}_{n,\mathrm{sup}}(\mathbb{R})$ et $A = QR$.

1) Écrire une fonction `DecQR` qui étant donnée A rend Q et R. Les calculs seront fait en réels flottants.

2) Appliquer cet algorithme à des matrices A prises au hasard. On vérifiera bien que $A = QR$, que ${}^tQQ$ est l'identité et que $R \in \mathcal{T}^{++}_{n,\mathrm{sup}}(\mathbb{R})$.

3) Écrire (ou retrouver dans un ancien fichier) une fonction `Triangle` qui résout un système triangulaire $RX = B$.

4) Écrire une fonction qui utilise `DecQR` et `Triangle` pour résoudre une équation $AX = B$ où $A \in \mathrm{GL}_n(\mathbb{R})$ et $B \in M_{n,1}(\mathbb{R})$.

Remarque. Pour cette décomposition QR, on peut aussi utiliser une autre méthode (plus stable numériquement) appelée méthode de Householder. Cette méthode permet aussi de calculer la décomposition QR de matrices rectangulaires et peut s'appliquer à la résolution du problème des moindres carrés.

Nous ne le ferons pas ici.