TP2 : La décomposition QR et les problèmes de moindres carrés

Vous êtes invités à créer dans un répertoire `.../TPM55/QRetMC/` le fichier `scriptTP2.sce` pour les exercices ci-dessous (comportant les lignes de code correspondant à la résolution de l'exercice) et le fichier `fonctionsTP2.sci` pour les algorithmes dont on a besoin (comportant les fonctions).

En cas de blocage, commencez toujours par regarder l'aide !!

1. Gram-Schmidt et Gram-Schmidt modifié

1.1. Programmer une fonction `[Q,R]=GS(A)` qui renvoie les matrices Q et R dans la décomposition QR économique de la matrice $A \in \mathcal{M}_{mn}(\mathbb{K})$, avec $\mathrm{rg}(A) = n \leq m$, par l'algorithme de Gram-Schmidt (voir point 3.2.1 du cours). On utilisera la vectorisation pour calculer les vecteurs colonnes $R(1 : k-1, k)$, $k = 2, \ldots, n$.

1.2. Tester la fonction `GS` pour les matrices

$$A_1 = \begin{pmatrix} 10 & -1 & 4 \\ 2 & 5 & -2 \\ -5 & 1 & 7 \end{pmatrix} \quad \text{et} \quad A_2 = \begin{pmatrix} & A_1 & \\ 0 & 4 & 3 \\ 6 & -3 & 9 \end{pmatrix}.$$

Calculer $\| Q_j R_j - A_j \|_2$ et $\| Q_j^H Q_j - I_3 \|_2$ pour $j = 1, 2$, où I_3 est la matrice identité de taille 3.

1.3. Pour $n = 2, \ldots, 10$ et $j = 1, \ldots, n$, soient $x_{nj} = -1 + \frac{2(j-1)}{n-1}$ des points équidistants dans l'intervalle $[-1, 1]$, et $V_n \in \mathcal{M}_n(\mathbb{R})$ la matrice de Vandermonde définie par

$$V_n = \begin{pmatrix} 1 & x_{n1}^1 & \ldots & x_{n1}^{n-1} \\ 1 & x_{n2}^1 & \ldots & x_{n2}^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{nn}^1 & \ldots & x_{nn}^{n-1} \end{pmatrix}.$$

Pour chaque n, calculer la décomposition $V_n = Q_n R_n$ à l'aide de la fonction `GS`. Tracer le graphe de $\| Q_n^H Q_n - I_n \|_2$ en fonction de n. On utilisera l'axe des ordonnées en échelle logarithmique. Comparer avec les graphes de $\mathrm{cond}_2(R_n)$ et $\mathrm{cond}_2(R_n)^2$.

1.4. Programmer une fonction `[Q,R]=GSmod(A)` qui renvoie les matrices Q et R dans la décomposition QR économique de la matrice $A \in \mathcal{M}_{mn}(\mathbb{K})$ par l'algorithme de

Gram-Schmidt modifié (voir point 3.2.2 du cours). On utilisera la vectorisation pour calculer les vecteurs lignes $R(k, k+1:n)$, $k = 1, \ldots, n-1$.

1.5. Tester la fonction `GSmod` pour les matrices de question 1.2. et 1.3. Comparer, à travers d'un graphe, les résultats obtenus (notamment en perte d'orthogonalité du facteur Q) avec ceux obtenus par la fonction `GS` et par la commande `qr` de Scilab qui renvoie une décomposition QR pleine. Par conséquent, si les matrices QM et RM sont celles obtenues par la commande qr de Scilab, on devrait s'intéresser aux sous-matrices $QM(1:m, 1:n)$ et $RM(1:n, 1:n)$.

2. Householder

2.1. Soit donné un vecteur $v \in \mathbb{R}^k$ où $k > 1$. Tester la commande `householder` de Scilab pour calculer le vecteur unitaire $u \in \mathbb{R}^k$ tel que

$$(I_k - 2uu^T)v = \pm \parallel v \parallel_2 e_1^{(k)}, \tag{1}$$

où $e_1^{(k)} = (1, 0, \ldots, 0)^T \in \mathbb{R}^k$ et le signe $\pm$ déterminé par les critères de stabilité vus en cours. Calculer u pour le vecteur $v = (1, 1, 1, 1)^T$ et vérifier l'égalité en (1).

2.2. Programmer une fonction `[Q,R]=HH(A)` qui renvoie les matrices Q et R dans la décomposition QR de la matrice $A \in \mathcal{M}_{mn}(\mathbb{R})$ par la méthode de Householder ; Pour la mise à jour de Q, on calculera le produit $Q(I - 2uu^*)$ en complexité $\mathcal{O}(m^2)$ en formant le vecteur Qu.

2.3. Tester la fonction `HH` pour les matrices de question 1.2. et 1.3. Comparer les résultats obtenus avec ceux obtenus par la commande `qr` de Scilab.

3. Les problèmes de moindres carrés

Soit donné un nuage de points $\{(x_k, y_k, z_k)\}_{k=1}^m \subset \mathbb{R}^3$. Nous souhaitons trouver un plan sous la forme $z = h(x, y) = n_0 + n_1 x + n_2 y$ de sorte que pour les points de notre nuage, la distance $\sum_{k=1}^m |z_k - h(x_k, y_k)|^2$ soit minimum.

3.1. Vérifier qu'il s'agit d'un problème aux moindres carrés – c'est-à-dire, exprimer A, b et x en fonction des données et des inconnues.

3.2. Programmer une fonction `[x]=MC(A,b)` qui, à l'aide d'une décomposition QR par la méthode de Householder, renvoie la solution `x` du problème

$$\min\{\parallel b - Ax \parallel : \ x \in \mathbb{R}^n\}$$

(voir point 3.1.6 du cours). On utilisera l'algorithme de remontée du TP1 pour résoudre le système d'équations

$$Rx = Q^T b.$$

3.3. Étant donnés le nuage de points $\{(2,2,1),(3,3,2),(1,4,3),(5,6,4),(0,3,5)\}$, appliquer la fonction `MC` pour calculer la solution de $Ax = b$ au sens des moindres carrés. Comparer le résultat obtenu avec celui obtenu par la commande `pinv(A)*b` de Scilab.

3.4. Tracer le plan $z = h(x,y)$ sur $[0,6] \times [0,6]$ et les segments de projection

$$[(x_k, y_k, z_k),\ (x_k, y_k, h(x_k, y_k))], \quad k = 1, \ldots, m.$$

Indication : la commande

```
[X,Y]=meshgrid(0:.1:6,0:.1:6)
```

crée un maillage de $[0,6] \times [0,6]$ et pour tracer le plan $z = y - x$ on fera

```
mesh(X,Y,X-Y)
```

On pourra ensuite tracer un segment de projection $[(a_1, a_2, a_3),\ (b_1, b_2, b_3)]$ en utilisant les commandes suivantes :

```
param3d([a_1,b_1],[a_2,b_2],[a_3,b_3]);
p=get("hdl");
p.polyline_style=4;
p.thickness=3;
p.arrow_size_factor=1;
```