TP 6 : Procédé d'orthonormalisation de Gram-Schmidt

Dans ce TP on met en oeuvre la méthode d'orthonormalisation de Gram-Schmidt originale et sa version modifiée, plus stable numériquement. On illustre l'instabilité de la méthode originale sur un exemple simple.

Le procédé de Gram-Schmidt est une méthode pour orthonormaliser une famille libre de vecteurs d'un espace vectoriel muni d'un produit scalaire. A partir d'une famille libre $(v_1, ..., v_n)$ on construit une famille orthonormale $(e_1, ..., e_n)$ qui engendre les mêmes espaces vectoriels successifs : pour tout j inférieur à n, $F_j = \text{Vect}(e_1, \ldots, e_j) = \text{Vect}(v_1, \ldots, v_j)$.

L'étape générale de l'algorithme consiste à soustraire au vecteur v_{j+1} sa projection orthogonale sur l'espace F_j. On s'appuie sur la famille orthonormale déjà construite pour le calcul de projection. Notons le projecteur sur la direction $\mathbf{u}$ par

$$\text{proj}_{\mathbf{u}}\,\mathbf{v} = \frac{\langle \mathbf{u}, \mathbf{v}\rangle}{\langle \mathbf{u}, \mathbf{u}\rangle}\mathbf{u}.$$

L'algorithme s'écrit :

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1, & \mathbf{e}_1 &= \frac{\mathbf{u}_1}{||\mathbf{u}_1||}\\
\mathbf{u}_2 &= \mathbf{v}_2 - \text{proj}_{\mathbf{u}_1}\,\mathbf{v}_2, & \mathbf{e}_2 &= \frac{\mathbf{u}_2}{||\mathbf{u}_2||}\\
\mathbf{u}_3 &= \mathbf{v}_3 - \text{proj}_{\mathbf{u}_1}\,\mathbf{v}_3 - \text{proj}_{\mathbf{u}_2}\,\mathbf{v}_3, & \mathbf{e}_3 &= \frac{\mathbf{u}_3}{||\mathbf{u}_3||}\\
&\vdots \quad \vdots & &\vdots\\
\mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{u}_j}\,\mathbf{v}_k, & \mathbf{e}_k &= \frac{\mathbf{u}_k}{||\mathbf{u}_k||}
\end{aligned}$$

1. Implémenter ce procédé en créant une fonction `gramschmidt` prenant une matrice rectangulaire en entrée et, si la famille de vecteurs colonnes est libre, renvoyant une matrice de même taille avec les vecteurs orthonormaux issus du procédé ci-dessus. On veillera à tester si la matrice a au moins autant de lignes que de colonnes, et à afficher un message d'erreur si la famille n'est pas libre, c'est à dire si la norme d'un vecteur devient trop petite ($< 10^{-20}$ par exemple). Application : tester la procédure sur la matrice

$$A = \begin{pmatrix} 1 & 2 & 3\\ 2 & -1 & 2\\ 1 & 1 & 2 \end{pmatrix}$$

2. Cette méthode originelle est instable numériquement. Pour l'illustrer, considérer le cas

$$A = \begin{pmatrix} 1 & 1 & 1\\ \varepsilon & 0 & 0\\ 0 & \varepsilon & 0\\ 0 & 0 & \varepsilon \end{pmatrix}$$

avec par exemple $\varepsilon = 10^{-10}$. Quel est le résultat exact ? Quel résultat donne votre procédure d'orthonormalisation ? Est-ce une famille orthogonale ? Cette anomalie vient du fait que du fait des erreurs d'arrondi, les vecteurs $\mathbf{u}_k$ ne sont pas orthogonaux. Cependant le procédé peut être stabilisé en l'implémentant de la manière suivante : Plutôt que de calculer $\mathbf{u}_k$ par $\mathbf{u}_k = \mathbf{v}_k - \text{proj}_{\mathbf{u}_1}\,\mathbf{v}_k - \text{proj}_{\mathbf{u}_2}\,\mathbf{v}_k - \cdots - \text{proj}_{\mathbf{u}_{k-1}}\,\mathbf{v}_k$, on le calcule par

$$\begin{aligned}
\mathbf{u}_k^{(1)} &= \mathbf{v}_k - \text{proj}_{\mathbf{u}_1}\,\mathbf{v}_k,\\
\mathbf{u}_k^{(2)} &= \mathbf{u}_k^{(1)} - \text{proj}_{\mathbf{u}_2}\,\mathbf{u}_k^{(1)},\\
&\vdots\\
\mathbf{u}_k^{(k-2)} &= \mathbf{u}_k^{(k-3)} - \text{proj}_{\mathbf{u}_{k-2}}\,\mathbf{u}_k^{(k-3)},\\
\mathbf{u}_k &= \mathbf{u}_k^{(k-2)} - \text{proj}_{\mathbf{u}_{k-1}}\,\mathbf{u}_k^{(k-2)}.
\end{aligned}$$

Ce calcul donnerait le même résultat en arithmétique exacte mais est plus stable en arithmétique approchée.

3. Implémenter ce second algorithme en modifiant très légèrement la fonction précédente et comparer le résultat qu'il fournit sur l'exemple ci-dessus. Etonnant, non ?