TP2: Résolution de systèmes linéaires

October 28, 2016

Objectifs de la séance

1. Méthodes de résolution de systèmes linéaires
2. Utilisation de Python: utilisation de numpy/scipy pour l'algèbre linéaire

1 Commandes Python pour l'algèbre linéaire

Ouvrez un navigateur de fichier dans votre répertoire personnel. Créez un répertoire AN_TP (si ce n'est pas déjà fait) puis créez un dossier "TP2_NOM". Lancez Python et sélectionnez "TP2_NOM" en tant que répertoire courant.

Soit $H \in M_5(\mathbb{R})$ la matrice

$$H = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}.$$

1. Créez un fichier mat-hilbert.py où vous écrirez vos commandes PYTHON
2. Rentrez la matrice H
3. Calculez la norme de H subordonnée à la norme 1, la norme 2 et la norme ∞
4. Calculez le conditionnement de H pour les 3 normes données précédemment
5. Calculer le déterminant de H
6. Calculez les valeurs propres et vecteurs propres de H
7. Soit $b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ et $\delta b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10^{-2} \end{pmatrix}$. Calculer les solutions x_0 et x_1 de $H\,x_0 = b$ et $H\,x_1 = b + \delta b$.
8. Calculer l'erreur relative $e = |x_1 - x_0|/|x_0|$ et expliquer le résultat obtenu.

2 Résolution d'un système linéaire "bien conditionné"

On s'intéresse à la résolution du système linéaire $A\,x = b$, où $A \in M_n(\mathbb{R})$ est la matrice définie par

$$A = \begin{pmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 4 & 1 \\ 0 & \dots & 0 & 1 & 4 \end{pmatrix}$$

Cette matrice intervient dans la détermination d'une spline (fonction polynomiale par morceaux) d'interpolation. Cette matrice est symétrique et elle est "creuse" (elle contient beaucoup de zéros. On verra que cela a des conséquences sur les couts de calcul. Dans la suite du TP, on présente plusieurs méthodes de résolution de $A\,x = b$.

1. Créez un fichier methode-direct.py
2. Ecrire un script permettant de construire la matrice A pour toute valeur de $n \in \mathbb{N}$
3. Pour $n = 10$, déterminer la factorisation LU à l'aide de la commande PYTHON dédiée (on aura pris soin d'importer les modules adéquates). Que remarque-t-on sur les matrices L et U?
4. Montrer que le nombre d'opérations pour faire cette factorisation $L\,U$ est $\mathcal{O}(n)$.
5. Pour $n = 100$, résoudre $A\,x_0 = b$ avec $b \in \mathbb{R}^n$ tel que $b_i = 1$, $\forall i$
6. Même question avec $A\,x_1 = b + \delta b$ et $\delta b_i = 10^{-2}\,\delta_{1i}$. Calculer l'erreur relative $|x_1 - x_0|/|x_0|$
7. En déduire un ordre de grandeur du conditionnement de la matrice A
8. Calculer directement le conditionnement de A directement avec la commande PYTHON dédiée (via les modules adéquates).
9. On s'intéresse maintenant à l'effet d'une perturbation sur la matrice A: refaire les questions 5 et 6 en calculant les solutions x_1 et x_2 de $(A + \delta A)x_1 = b$ et $(A + deltaB)x_2 = b$ avec $\delta A_{ij} = h\,(-1)^i\,\delta_{ij}$ et $\delta B_{ij} = h\,(-1)^i\,\delta_{j=i-1}$ pour différentes valeurs de $h = 0.5,\ 0.1,\ 0.01$. Refaire le calcul d'erreurs relatives et expliquer le résultat obtenu.
10. Reprendre les questions $5, 6, 8$ en faisant varier cette fois la taille de la matrice A (on prendra $N = 500$, $N = 1000$, $N = 2000$).
11. La matrice A est symétrique: en donner sa décomposition de Cholesky (on commencera par un cas avec n petit pour avoir une intuition du résultat)

3 Cas d'un système dont le conditionnement dépend de la taille de la matrice

Reprendre les questions des sections précédentes avec la matrice B donnée par

$$B = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

En particulier, faites varier la taille de la matrice B et observez l'évolution du conditionnement (prendre des très grandes valeurs de $N = 2000, 5000, 10000$).