Université de Marseille
Licence de Mathématiques, analyse numérique
Travaux Pratiques 2, en python

Exercice 1 (Etude d'un système particulier)
On s'intéresse au système $Ax = b$ avec

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \text{ et } b = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}.$$

1. Calculer les valeurs propres de A.

   [On pourra utiliser la fonction **numpy.linalg.eigvals**].

   En déduire que A est une matrice s.d.p..

2. Calculer le conditionnement de A en utilisant les valeurs propres de A. Comparer avec le résultat donné par la fonction **numpy.linalg.cond**.

3. Calculer le solution du système linéaire $Ax = b$ en utilisant la fonction **numpy.linalg.solve**.

4. Calculer le solution du système linéaire $Ax = b$ en utilisant le programme d'échelonnement du tp1 (qui, ici, consiste à utiliser la méthode de Gauss avec pivot partiel). Comparer avec la solution de la question précédente.

5. On perturbe maintenant légèrement le système en remplaçant b par $b + \delta_b$, avec $\delta_b = \begin{bmatrix} 0.1 \\ -0.1 \\ 0.1 \\ -0.1 \end{bmatrix}$. Calculer la nouvelle solution du système, notée $x + \delta_x$.

   Vérifier que $\dfrac{|\delta_x|}{|x|} \leq cond(A)\dfrac{|\delta_b|}{|b|}$.

On rappelle que la norme euclidienne s'obtient avec **numpy.linalg.norm**

Exercice 2 (L'inégalité sur le conditionnement est optimale)
Dans cet exercice, on construit un exemple pour lequel $\dfrac{|\delta_x|}{|x|} \leq cond(A)\dfrac{|\delta_b|}{|b|}$ avec $A \in \mathcal{M}_n(\mathbb{R})$ inversible. On pourra prendre, par exemple, $n = 10$.

1. Construire une matrice diagonale $D = diag(d_1, \ldots, d_nn)$ en choisissant d_1, $\ldots$, d_n de manière aléatoire entre 1 et 10.

   [utiliser **numpy.random.random** qui donne n nombres aléatoires compris entre 0 et 1.]

   Ordonner les nombres pour que $0 < d_1 < \ldots < d_n$ (utiliser **numpy.sort**). Enfin, utiliser **numpy.diag** pour construire D.

2. Construire une matrice Q (de $\mathcal{M}_n(\mathbb{R})$) orthogonale de la manière suivante : Construire une première matrice P avec des coefficients alétaoires (en général P est inversible, mais si P n'est pas inversible, on recommence le choix des coefficients). On obtient alors une matrice orthogonale Q en effectuant la décomposition QR de cette matrice (utiliser **numpy.linalg.qr**).

On choisit maintenant $A = QDQ^t$, avec D et Q données par les questions précédentes. La matrice A ainsi construite est symétrique définie positive. On remarque que $c_1(Q)$ est le vecteur propre associé à la plus petite valeur propre de A alors que $c_n(Q)$ est le vecteur propre associé à la plus grande valeur propre de A.

3. On pose $b = c_n(Q)$ et $\delta_b = c_1(Q)$ Calculer x et $x + \delta_x$ (tels que $Ax = b$ et $A(\delta_x) = \delta_b$) et vérifier que

$$\frac{|\delta_x|}{|x|} = cond(A)\frac{|\delta_b|}{|b|}.$$

Pour cet exemple le conditionnement mesure exactement le rapport entre les erreurs relatives. Mais, on remarque que cette égalité est obtenue pour des choix très particuliers de b et δ_b. L'exercice suivant montre que la sensibilité de la solution aux erreurs sur le second membre peut être bien meilleure que celle prédite par le conditionnement lorsque le problème provient d'une discrétisation d'une équation différentielle (ou, plus généralement, d'une équation aux dérivées partielles).

Exercice 3 (Conditionnement "efficace")

On s'intéresse, dans cet exercice, à la matrice de l'exercice 1 du tp1. Pour $n \in \mathbb{N}^\star$, on note A_n cette matrice (qui était notée A_h au tp1, avec $h = 1/(n+1)$). On considère le même problème que dans le tp1, c'est à dire celui correspondant à $f(x) = \pi^2 \sin(\pi x)$ pour $x \in [0, 1]$.
Pour $n \in \mathbb{N}^\star$ le problème discrétisé s'écrit donc $A_n x_n = b_n$.

1. Pour n variant entre 100 et 1000 :
   (a) Calculer b_n et x_n,
   (b) Choisir de manière aléatoire un vecteur de $\mathbb{R}^n$, noté δ_{b_n}, prenant ses valeurs entre 0 et 0.1. Calculer $x_n + \delta_{x_n}$ solution de $A_n(x_n + \delta_{x_n}) = b_n + \delta_{b_n}$ et calculer le nombre $cond_f(A_n)$ vérifiant $\dfrac{|\delta_{x_n}|}{|x_n|} = cond_f(A_n)\dfrac{|\delta_{b_n}|}{|b_n|}$.

2. Dessiner les graphes (Pour n variant entre 100 et 1000) des applications $n \mapsto cond(A_n)$ et $n \mapsto cond_f(A_n)$. Remarquer que contrairement au conditionnement de A_n, le rapport entre les erreurs relatives sur b et x (notée $cond_f(A_n)$) ne croit pas comme n^2 (mais reste borné).

   [Utiliser **matplotlib.pyplot.plot** et **matplotlib.pyplot.show**.]