Norme d’une matrice de Hankel infinie

ANDRINAJORO Mija Niaina

May 3, 2025

1. Resumé

L'objectif de cette étude est d'approcher numériquement la norme spectrale d'une des matrices de Hankel : matrice de Hilbert . Sur ce , on a essayé d'implémenter en python deux méthodes : méthode de la puissance et celui d' Arnoldi. On a trouvé que la norme spectrale converge vers un certain nombre quand la taille de la matrice tend vers $+\infty$.

2. Introduction

2.1. Contexte scientifique :

Une **matrice de Hankel** est une matrice carré dont les indices vérifient la relation :

$$a_{i,j} = a_{i-1,j-1}.$$

Une **matrice de Hilbert** est une matrice de Hankel particulière. Elle vérifie la relation :

$$a_{i,j} = \frac{1}{i+j+1}.$$

Les matrices de Hilbert sont connues pour être mal conditionnées c-à-d une petite variation de ses éléments peut provoquer des grandes variations dans la solution d'un système linéaire associé . En d'autres termes , une petite perturbation dans les données provoquent une erreur très importante dans le résultat calculé. Pour quantifier la sensibilité de la solution d'un système linéaire aux perturbations de données , on utilise la norme spectrale . Cette quantité est souvent appelée conditionnement .
Ainsi il est important de connaître les propriétés de la norme spectrale d'une matrice de Hilbert.

2.1. But :

Notre but est de prouver la convergence de la norme formuler une conjecture numérique pour la valeur limite .

2.3. Plan :

Voici les étapes qu'on a suivi :

❶ Calculer la norme pour les coupes n$\times$n $\leq$ 1200 par la méthode des puissances .

❷ Calculer la norme par la méthode d'Arnoldi .

❸ Conjecturer la convergence .

❹ Utiliser mpmath à haute précision sur n=100 pour valider la conjecture .

❺ Représentation graphique et conjecture numérique de la valeur limite .

3. Méthodes :

3.1 Génération d'une matrice de Hilbert :

On peut construire une matrice de Hilbert de taille n de façon récursive :

```
import numpy as np
def matrice_hilbert(n):
    M=np.zeros((n,n))##matrice nulle
    for i in range(n):
        for j in range(n):
            M[i,j]=1/(i+j+1)
    return M
```

On peut aussi obtenir une matrice de Hilbert en important le module **scipy** :

```
from scipy.linalg import hilbert
H=hilbert(n)
```

3.2. Calcul de la norme par méthode des puissances .

Norme spectrale : La norme spectrale d'une matrice M est la plus grande valeur singulière de M . Elle est définie par :

$$||M||_2 = \max_{||x||_2=1} ||Mx||_2.$$

Ici , $||x||_2$ désigne la norme euclidienne du vecteur x . Comme une matrice de Hilbert est symétrique , ses valeurs singulières coïncident avec les modules de ses valeurs propres .

Principe de la méthode de la puissance :

- Choisir un vecteur x_0 tel que Mx non nul.
- $x_{k+1} = Mx_k/||Mx_k||$.
- La valeur propre de plus grande module est la limite de (x_k)

Code python : En appliquant la methode de la puissance à M^tM , on obtient la plus grande valeur propre de M.

```
def     puissance_norme(M, i=10000, t=1e-10):
        n=M.shape[0] #taille de la matrice M
        x=np.random.rand(n) #x est le vecteur aleatoire initial
        x=x/np.linalg.norm(x)
        a=0
        for k in range(i):
              Mx=M@x
              x=Mx/np.linalg.norm(Mx)
              valeur_propre=x.T@M@x
              if abs(valeur_propre-a)<t:
                      return valeur_propre,k+1
              a=valeur_propre

        return valeur_propre
```

3.3. Accélération avec Arnoldi :

Principe :

- On construit une base orthonormale $\{q_1, ..., q_m\}$ du sous-espace

$$K_m(M, q_1) = span\{q_1, Mq_1, M^2q_1, M^{m-1}q_1\}.$$

- A chaque étape ,on calcule Mq_k puis on orthogonalise par rapport a la base existante $\{q_1, ..., q_k\}$ pour obtenirq_{k+1}.
- On obtient une matrice H_m de taille m×m de Hessenberg dont les valeurs propres sont des approximations des valeurs propres de M.

Code python : On a utilisé la fonction **eigsh** qui retourne les valeurs propres et d'une matrice.Elle utilise la méthode d'Arnoldi.

```
from scipy.sparse.linalg import eigsh
def arnoldi_norme(n):
     H = hilbert(n)
    valeur_propre,_ = eigsh(H, k=1, which='LM')
    return valeur_propre[0]
```

which='LM' signifie la valeur absolue est maximale.

3.4. Conjecturer la valeur limite

On a ajusté la courbe des résultats obtenu en utilisant une extrapolation et la fonction **scipy.optimize.curve-fit**

4. Analyse des résultats :

Les résultats obtenus sont illustrés dans les fichiers csv.

4.1. Comparaison entre methode de la puissance vs Arnoldi

La méthode d'Arnoldi est plus rapide surtout pour les grandes valeurs de N . Par exemple pour N=1000 le rapport entre temps de calcul via méthode de la puissance vs Arnoldi est d'environ 1.5 .

4.2. Etude de la convergence :

D'après les résultats $|||A||_N - ||A||_{N+100}|$ est au voisinage de 0 et diminue avec N . Donc , on peut dire que $(||A||_N)$ est de Cauchy . C-à-d elle converge vers une limite l.

4.3. Calcul à haute précision :

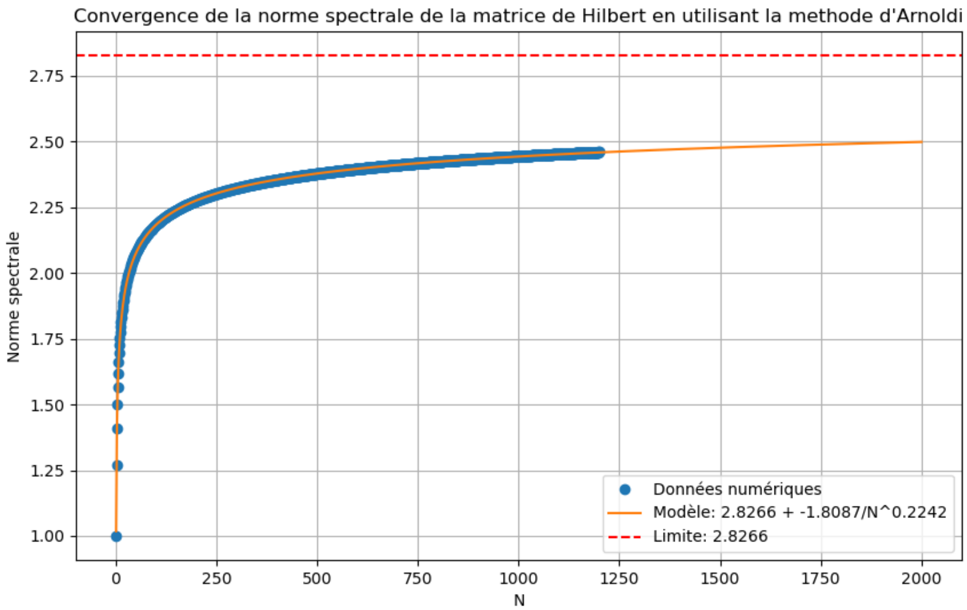
En observant les résultats , il semble que la limite $l \leq 2$. Ceci s'est confirmé quand on a calculé avec haute précision , pour N=100 : on obtient 2.18 avec 60 itérations.

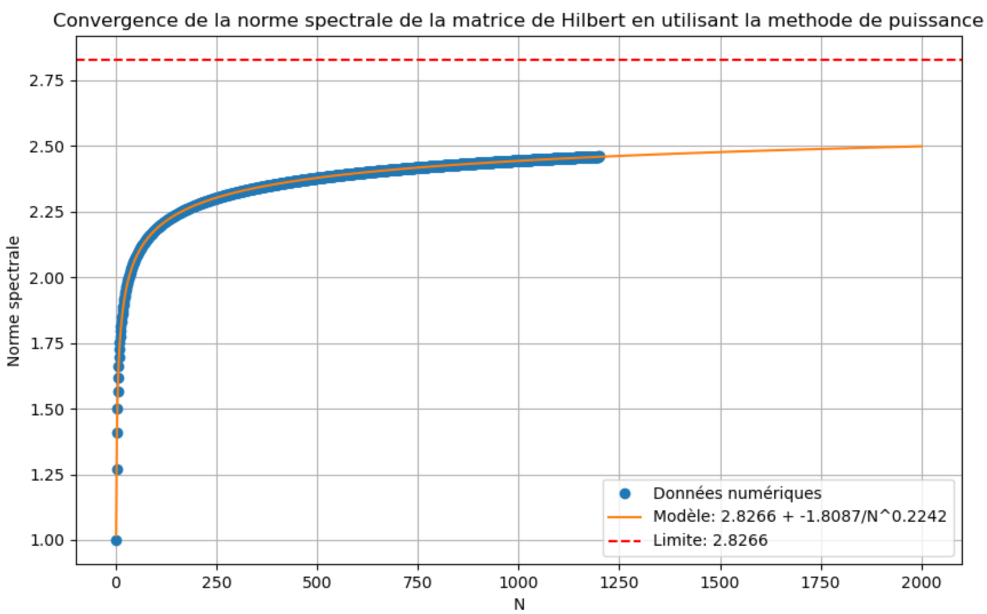
4.4 Valeur limite :

La norme spectrale converge vers 2.8 quelque soit la méthode utilisée .(voir les figures à la dernière page.)

5. Conclusion :

A travers cette étude ,on a montré que la norme spectrale d'une matrice de Hilbert converge vers une limite l qu'on a estimé numériquement =2.8 qui est assez proche de π (limite en théorie). L'utilisation de divers méthodes nous a permis de confirmer cette estimation.


Convergence de la norme spectrale de la matrice de Hilbert en utilisant la methode d'Arnoldi
Norme spectrale
N
Données numériques
Modèle: 2.8266 + -1.8087/N^0.2242
Limite: 2.8266


Convergence de la norme spectrale de la matrice de Hilbert en utilisant la methode de puissance
Norme spectrale
N
Données numériques
Modèle: 2.8266 + -1.8087/N^0.2242
Limite: 2.8266