Analyse Nmérique TP2

DUQUENOY Cyrile, CALLEA Angelo *

Université d'Aix-Marseille
2021-2022

L3 de Mathématiques
Second semestre

TP3, Analyse Numérique : Normes et Conditionnement

Abstract

Le but de ce TP est de mettre en pratique les différentes méthodes itératives et recherches de valeurs propres vus en cours.

On verra notamment la méthode de Jacobi, de Newton, de la Puissance, et méthode QR.

1 Exercice 3 : Méthode par la décomposition QR

1.1 Rappel

Toute matrice $A \in \mathbb{M}_n(\mathbb{R})$ peut se décomposer comme produit de deux matrice Q, R tel que Q soit orthogonale et R triangulaire supérieure à coefficients diagonaux positifs ou nuls. Dans le cas où A est inversible alors cette décomposition est unique.

1.2 Algorithme de la méthode QR

On se donne une méthode itérative comme ce qui suit :

Soit une matrice inversible A. On note $A_{(0)} = A$, qu'on va décomposer de la forme $A_{(0)} = Q_0 R_0$. Puis pour tout $k \in \mathbb{N}$, $A_{(k+1)} = R_k Q_k$.

Pour la plupart des matrices, cet algorithme permet d'approcher les valeurs propres de A. En effet, les coefficients diagonaux de A_k tendent vers les valeurs propres de la matrice

*Université Aix Marseille

A. De plus si A est symétrique, les colonnes de Q_k tendent vers les vecteurs propres associés.

Bien que la démonstration ne soit pas faites dans le cours, on va ici mettre en application l'algorithme QR par le calcul numérique.

1.3 Application Numérique

Soit A une matrice inversible et $\epsilon > 0$ tel que l'algorithme donné ce-dessus se stoppe quand tout les coefficients sous la diagonale de A^k sont inférieurs à ϵ. Il se peut que la méthode converge trop lentement voire même pas du tout alors on mets aussi une clause d'itération maximum donné à au plus 1000 itérations.

On construit ensuite une matrice $M \in \mathbb{M}_4(\mathbb{R})$ aléatoire et on pose $P = M^t M$. SI P n'est pas inversible alors on recommence jusqu'à ce qu'elle le soit.
On construit alors $A_i = PD_iP^{-1}$, $i = 1, 2, 3$ avec $D_1 = diag(6, 5, 2, 1)$, $D_2 = diag(5, 1, 2, 6)$, $D_3 = diag(5, 3, 2, 3)$.

On peut alors estimer les valeurs propres des A_i, $i = 1, 2, 3$ en utilisant la méthode QR décrit en début de section.

En effet, les valeurs approchés des valeurs propres sont très proches voire exactement les mêmes que celles qu'on trouve en utilisant numpy.linalg.eigenval.
Notre algorithme fonctionne donc comme il le faut.

1.3.1 Exemple de non convergence

On se donne ici une matrice $A_4 = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

On remarque alors que la fonction construite sur python tourne en boucle, en fait la méthode ne converge pas.

2 Méthode du Point Fixe

On se donne une fonction $g(x) = x^2 - 3$ et on pose $f(x) = x - g(x)$.
On mets en place l'algorithme du point fixe appliqué à f pour un $x_0 = 1$.
On remarque que python mets de plus en plus de temps à calculer les itérations, se sont de

trop grands nombres.

En fait, la méthode ne converge pas. Il est assez facile de voir que f n'est pas contractante sur $\mathbb{R}$. En effet, par la définition, si f était contractante alors pour tout $x, y \in \mathbb{R}$, $\|f(x) - f(y)\| <= K\|x - y\|$ avec $K \in]0, 1[$. Or il suffit de remarquer que $\|f(x_3) - f(x_2)\| = 78 > \|x_3 - x_2\| = 6$, il n'existe donc pas un tel K qui vérifie f contractante. La méthode du point fixe ne peut pas s'appliquer.