Université Paris I, Panthéon - Sorbonne

Deuxième Année Licence M.I.A.S.H.S. 2016 – 2017

TP de Méthodes Numériques n^0 10 : Calcul et utilisation de valeurs propres

On continue de s'intéresser dans ce TP aux applications possible en algèbre linéaire d'un logiciel numérique comme R. Nous nous centrons ici sur les notions de valeurs et vecteurs propres, la notion de diagonalisation et les utilisations possibles de ces notions.

Un algorithme d'obtention de la valeur propre de plus grand module

On suppose ici que l'on dispose d'une matrice carrée A de taille n et dont les valeurs propres, au nombre de $1 \leq p \leq n$ et pouvant être complexes, sont $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_p|$. Voici un algorithme permettant d'approcher d'aussi près que l'on désire λ_1:

```
n=ncol(A)
tol=1e-06
nmax=1000
x0=matrix(1,n,1)
x0=x0/norm(x0)
n=1
pro=A%*%x0
lambda=t(x0)%*%pro
while (err>tol & n<=nmax)
{n=n+1
x=pro
x=x/norm(x)
pro=A%*%x
lambda[n]=t(x)%*%pro
err=abs(lambda[n]-lambda[n-1])/abs(lambda[n])}
plot(lambda)
```

Appliquer la méthode à une matrice de taille 2, par exemple $A = \begin{pmatrix} 3 & 0 \\ 4 & -4 \end{pmatrix}$. Déterminer les valeurs propres de A et vérifier que `lambda`, calculée avec l'algorithme, est proche de la valeur propre de plus grande valeur absolue. Comparer également la dernière valeur de x avec un vecteur propre normé associé à la valeur propre de plus grande valeur absolue. Qu'en pensez-vous?
Plus généralement, à partir de l'algorithme, écrire le vecteur `x` et le réel `lambda` à l'ordre n en fonction de puissances de A. Si on note v_1 un vecteur propre normé associé à la valeur propre λ_1, que peut-on dire de ${}^t v_1 A v_1$? Dans le cas où A est déjà une matrice diagonale, essayer de comprendre pourquoi `lambda[n]` converge vers λ_1 et pourquoi il est nécessaire que $|\lambda_1|$ soit strictement plus grand que $|\lambda_2|$.
Appliquer l'algorithme à des matrices symétrique A de tailles $n = 2$, $n = 3$, $n = 10$ et $n = 1000$ en vérifiant à chaque fois que vous retrouvez bien les résultats obtenus avec la commande `eigen` de R.
Que se passe-t-il si A est une matrice réelle dont la valeur propre de plus grand module est un nombre complexe? Faire un essai avec une matrice de taille 2.

Calcul de toutes les valeurs et tous les vecteurs propres d'une matrice

On va ici directement utiliser la décomposition QR d'une matrice. Elle est obtenue à partir des commandes suivantes. Notons que ces commandes nécessitent l'installation du package `pracma`, ce qui se fait dans l'onglet `Packages` et ceci après avoir choisi un site miroir de CRAN (par exemple celui de Paris 1).

```
library(pracma)
QR=gramSchmidt(A)
Q=QR$Q
R=QR$R
```

On pourra ensuite verifier que $Q\% * \%R = A$. Une fois ceci fait on peut mettre en place l'algorithme qui permettra d'approcher d'aussi près que l'on veut les différentes valeurs propres d'une matrice. En effet, on montre que l'on peut effectuer une suite de décompositions QR construite à partir de A de la manière suivante:

$$A_1 = A = Q_1 R_1, \qquad A_{k+1} = R_k Q_k \quad \text{pour } k \in \mathbf{N}^*.$$

Alors la suite de matrices (A_k) converge vers une matrice triangulaire supérieure dont la diagonale est constituée des valeurs propres de A lorsque celle-ci possède n valeurs propres simples distinctes et non nulles.
Programmer cette suite de décompositions. Vérifier son bon fonctionnement à partir d'exemples simples. Faire des essais lorsque la matrice A possède des valeurs propres doubles ou lorsque A n'est pas inversible.

Exercices

Exercice 1. Pour $n \in \mathbf{N}^*$, on considère une matrice de Vandermonde, c'est-à-dire que l'on choisit $(\alpha_1, \cdots, \alpha_n)$, n réels distincts, tels que $A = (\alpha_i^{j-1})_{1 \le i,j \le n}$. Déterminer numériquement les valeurs propres de A lorsque $\alpha_i = 1/i$ pour différentes valeurs de n. Tracer l'évolution de la plus grande valeur propre de A en fonction de n. Quel type d'évolution est-ce?

Exercice 2. Soit la matrice de Hilbert définie dans les TP précédents ($a_{ij} = (i + j - 1)^{-1}$ pour $1 \le i, j \le n$). Pour n variant de 1 à 20, déterminer numériquement la plus grande et la plus petite valeur propre, puis le rapport entre les 2. Quelle conclusion en tirez vous par rapport à la difficulté d'inverser la matrice?

Exercice 3. Générer une matrice carrée d'ordre (10×10) quelconque M (fonction *rand* ...) puis calculer son carré et sa puissance 100. On appelle λ_M la valeur propre de plus grand module de M (une telle matrice M est diagonalisable avec une probabilité en théorie égale à 1). Vérifier que si $|\lambda_M| > 1$ alors M^n à tendance à "exploser" (en quel sens?) lorsque $n \to \infty$, mais que si $|\lambda_M| \le 1$ (pour ce cas, il pourra être utile de générer d'abord les valeurs propres, puis la matrice M), M^n tend vers une matrice limite lorsque $n \to \infty$. Expliquer ceci.

Exercice 4. Soit une suite récurrente $(u_n)_n$ définie par $u_{n+p} + a_{p-1}\, u_{n+p-1} + \cdots + a_0\, u_n = 0$ pour $n \ge 0$, et $u_0 = u_1 = \cdots = u_{p-1} = 1$, les a_i étant des réels fixés et $p \ge 1$. Montrer que (u_n) est bien définie de manière unique. Soit le vecteur colonne $U_n = {}^t(u_{n+p}, u_{n+p-1}, \cdots, u_n)$. Montrer qu'il existe une matrice A carrée de taille p composée des a_i telle que $U_{n+1} = A\, U_n$. Pour $p = 10$, générer les a_i de manière aléatoire (commande `runif(10,-1,1)`), puis déterminer les u_i pour $i = 1, \cdots, n$ ($n = 1000$ pour commencer). Soit λ la valeur propre de plus grand module de A. Montrer numériquement que u_n/λ^n converge vers une constante. Expliquer théoriquement ce résultat.