Centre Universitaire de Mila
Département Mathématiques et Informatique

Chapitre 2: Approximation des valeurs propres et des vecteurs propres

Dr. Aissa Boulmerka
aissa.boulmerka@gmail.com

1- INTRODUCTION

Introduction

- Nous abordons dans ce chapitre l'approximation des **valeurs propres** et des **vecteurs propres** d'une matrice $A \in C^{n \times n}$.
- Il existe deux classes de méthodes numériques pour traiter ce problème :
  - **Méthodes partielles:** permettent le calcul approché des valeurs propres extrêmes de A (c'est-à-dire celles de plus grand et de plus petit module), e.g. la méthode de la puissance.
  - **Méthodes globales**: fournissent des approximations de tout le spectre de A, e.g. la méthode QR .
- Certaines méthodes de calcul des valeurs propres permettent également le calcul des vecteurs propres. Ainsi, la méthode de la puissance fournit l'approximation d'une paire particulière de valeur propre/vecteur propre.
- Mais les autre méthodes ne donnent pas systématiquement les vecteurs propres associés. Par exemple, la méthode QR permet le calcul de la forme de Schur réelle de A, c'est-à-dire une forme canonique qui contient toutes les valeurs propres de A, mais elle ne fournit aucun des vecteurs propres.

Introduction

Pour une matrice A réelle, carrée de $n \times n$ éléments, une **valeur propre** $\boldsymbol{\lambda}$ et un **vecteur propre** correspondant $x \neq 0$ satisfont $\mathbf{Ax} = \boldsymbol{\lambda}\mathbf{x}$.

Il existe n paires propres (éventuellement complexes) $\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_n}$ et vecteurs propres $x_1, x_2, \ldots, x_n$ s.t. $\mathbf{Ax_i} = \boldsymbol{\lambda_i}\mathbf{x_i}$. Pour $\boldsymbol{\Lambda} = \boldsymbol{diag}\,(\boldsymbol{\lambda i})$

$$\mathbf{AX} = \mathbf{X\Lambda}.$$

Si $\boldsymbol{A}$ n'est pas défectueux alors la matrice de vecteurs propres $\boldsymbol{X}$ est non singulière:

$$\mathbf{X^{-1}AX} = \boldsymbol{\Lambda}.$$

Pour une matrice $\mathbf{A}$ réelle, $\boldsymbol{m} \times \boldsymbol{n}$, **la décomposition en valeurs singulières (SVD)** est

$$\mathbf{A} = \mathbf{U\Sigma V^T}$$

où $\mathbf{U}$ $\boldsymbol{m} \times \boldsymbol{m}$ et $\mathbf{V}$ $\boldsymbol{n} \times \boldsymbol{n}$ sont des matrices orthogonales et $\boldsymbol{\Sigma}$ est une matrice diagonale "constitué de zéros et de valeurs singulières

$$\sigma_1 \geq \sigma_2, \cdots \geq \sigma_r > 0, \text{r} \leq \min(\text{m}, \text{n}).$$

Remarques:

- les valeurs singulières sont les racines carrées des valeurs propres de $\mathbf{A^T A}$
- Une matrice défectueuse est une matrice carrée qui n'a pas une base complète de vecteurs propres, et n'est donc pas diagonalisable.

2- MÉTHODE DE LA PUISSANCE

Méthode de la puissance

- La **méthode de la puissance** fournit une très bonne approximation des valeurs propres **extrémales** d'une matrice et des vecteurs propres associés.
- On notera $\boldsymbol{\lambda_1}$ et $\boldsymbol{\lambda_n}$ les valeurs propres ayant respectivement le plus grand et le plus petit module.
- La résolution d'un tel problème présente un grand intérêt dans beaucoup d'applications concrètes : le traitement d'image et la vision par ordinateur, sismique, étude des vibrations des structures et des machines, analyse de réseaux électriques, mécanique quantique, . . .
- dans lesquelles $\boldsymbol{\lambda_n}$et le vecteur propre associé $\mathbf{x}_{\boldsymbol{n}}$ permettent la détermination de la **fréquence propre** et du **mode fondamental** d'un système physique donné.

Méthode de la puissance

- Il peut être aussi utile de disposer d'approximations de $\boldsymbol{\lambda_1}$ et $\boldsymbol{\lambda_n}$ pour analyser des méthodes numériques.
- Par exemple, si $\mathbf{A}$ est symétrique définie positive, $\boldsymbol{\lambda_1}$ et $\boldsymbol{\lambda_n}$ permettent de calculer la valeur optimale du paramètre d'accélération de la méthode de Richardson, d'estimer son facteur de réduction d'erreur, et d'effectuer l'analyse de stabilité des méthodes de discrétisation des systèmes d'équations différentielles ordinaires.

Méthode de la puissance

- Soit $\mathrm{A} \in \mathbb{C}^{\mathrm{n}\times\mathrm{n}}$ une matrice diagonalisable et soit $\mathrm{X} \in \mathbb{C}^{\mathrm{n}\times\mathrm{n}}$ la matrice de ses vecteurs propres $\mathbf{x}_{\boldsymbol{i}}$, pour $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{n}$. Supposons les valeurs propres de A ordonnées de la façon suivante

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \ldots \geq |\lambda_n| \qquad \textbf{(1)}$$

- et supposons que $\boldsymbol{\lambda_1}$ ait une multiplicité algébrique égale à $\mathbf{1}$. Sous ces hypothèses, $\boldsymbol{\lambda_1}$ est appelée valeur propre **dominante** de la matrice A.
- Etant donné un vecteur initial arbitraire $\mathbf{q}^{(0)} \in \mathbb{C}^n$ de norme euclidienne égale à $\mathbf{1}$, considérons pour $\boldsymbol{k} = \mathbf{1}, \mathbf{2}, \ldots$ la méthode itérative suivante, connue sous le nom de **méthode de la puissance** :

$$\begin{aligned} \mathbf{z}^{(k)} &= A\mathbf{q}^{(k-1)}, \\ \mathbf{q}^{(k)} &= \frac{\mathbf{z}^{(k)}}{\left\|\mathbf{z}^{(k)}\right\|_2}, \\ v^{(k)} &= \left(\mathbf{q}^{(k)}\right) * A\mathbf{q}^{(k)}. \end{aligned} \qquad \textbf{(2)}$$

Méthode de la puissance

```
function [lambda,x,iter,relres] = powerm(A, z0, tol, nmax)
%POWERM Méthode de la puissance calcule la valeur propre LAMBDA de plus grand module de la matrice A et un vecteur propre correspondant X de norme un.
% TOL est la tolérance de la méthode.
% NMAX est le nombre maximum d'itérations.
% Z0 est la donnée initiale.
% ITER est l'itération à laquelle la solution X a été calculée.

q=z0/norm(z0); q2=q;
relres=tol+1; iter=0; z=A*q;
```

```
while relres(end)>=tol && iter<=nmax
    q=z/norm(z); z=A*q;
    lambda=q'*z; x=q;
    z2=q2'*A; q2=z2/norm(z2); q2=q2';
    y1=q2; costheta=abs(y1'*x);
    if costheta >= 5e-2
        iter=iter+1;
        temp=norm(z-lambda*q)/costheta;
        relres=[relres; temp];
    else
        fprintf('Valeur propre multiple'); break;
    end
end
return
```

Exemple: Méthode de la puissance

```
clear; clc;
A = [15 -2 2; 1 10 -3; -2 1 0];
z0 = [1; 1; 1];
tol = 10^-11;
nmax = 1000;
[lambda,x,iter,relres]=powerm(A,z0,tol,nmax);
fprintf('lambda=%.15f\n',lambda);
fprintf('iter=%d\n',iter);
```

lambda=14.102555760082943
iter=84

3- MÉTHODE DE LA PUISSANCE INVERSE

Méthode de la puissance inverse

- Nous recherchons une approximation de la valeur propre d'une matrice $\mathrm{A} \in \mathbb{C}^{\mathrm{n}\times\mathrm{n}}$ la **plus proche** d'un nombre $\boldsymbol{\mu} \in \mathbb{C}$ donné, avec $\boldsymbol{\mu} \notin \boldsymbol{\sigma}(\boldsymbol{A})$. On peut pour cela appliquer la méthode de la puissance **(2)** à la matrice $(\mathrm{M}_{\mu})^{-1} = (\mathrm{A} - \mu\mathrm{I})^{-1}$, ce qui conduit à la méthode des **itérations inverses** ou méthode de **la puissance inverse**. Le nombre $\boldsymbol{\mu}$ est appelé shift en anglais.
- Les valeurs propres de ${\mathrm{M}_{\mu}}^{-1}$sont $\boldsymbol{\xi_i} = (\boldsymbol{\lambda_i} - \boldsymbol{\mu})^{-\mathbf{1}}$; supposons qu'il existe un entier $\boldsymbol{m}$ tel que

$$|\lambda_m - \mu| < |\lambda_i - \mu|, \qquad \forall i = 1, \dots, n \text{ et } i \neq m. \qquad \textbf{(3)}$$

- Cela revient à supposer que la valeur propre $\boldsymbol{\lambda_m}$ qui est la plus proche de $\boldsymbol{\mu}$ a une multiplicité égale à $\mathbf{1}$. De plus, **(3)** montre que $\boldsymbol{\xi_m}$ est la valeur propre de ${\boldsymbol{M_{\mu}}}^{-\mathbf{1}}$ de plus grand module; en particulier, si $\boldsymbol{\mu} = \mathbf{0}$, $\boldsymbol{\lambda_{\mathrm{m}}}$ est la valeur propre de A de plus petit module.

Méthode de la puissance inverse

- Etant donné un vecteur initial arbitraire $\mathbf{q}^{(0)} \in \mathbb{C}^{n}$ de norme euclidienne égale à $\mathbf{1}$, on construit pour $\boldsymbol{k} = \mathbf{1}, \mathbf{2}, \ldots$ la suite définie par

$$(\mathrm{A} - \mu \mathrm{I})\mathbf{z}^{(k)} = \mathbf{q}^{(k-1)},$$

$$\mathbf{q}^{(k)} = \frac{\mathbf{z}^{(k)}}{\left\|\mathbf{z}^{(k)}\right\|_2}, \qquad \textbf{(4)}$$

$$\sigma^{(k)} = \left(\mathbf{q}^{(k)}\right) * \mathrm{A}\mathbf{q}^{(k)}.$$

- Remarquer que les vecteurs propres de M_{μ} sont les mêmes que ceux de A puisque $\mathrm{M}_{\mu} = \mathrm{X}(\Lambda - \mu \mathrm{I}_{\mathrm{n}})\mathrm{X}^{-1}$, où $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$. Pour cette raison, on calcule directement le quotient de Rayleigh dans **(4)** à partir de la matrice A (et non à partir de ${\mathrm{M}_{\mu}}^{-1}$).
- La différence principale par rapport à **(2)** est qu'on doit résoudre à chaque itération k un système linéaire de matrice $\mathrm{M}_{\mu} = \mathrm{A} - \mu \mathrm{I}$.

Méthode de la puissance inverse

```
function [sigma,x,iter,relres] = invpower(A,
z0, mu, tol, nmax)
%INVPOWER Méthode de la puissance
inverse calcule la valeur propre LAMBDA
de plus petit module de la matrice A et un
vecteur propre correspondant X de norme
un.
% TOL est la tolérance de la méthode.
% NMAX est le nombre maximum
d'itérations. X0 est la donnée initiale.
% MU est le shift. ITER est l'itération à
laquelle la solution X a été calculée.
M=A-mu*eye(size(A)); [L,U,P]=lu(M);
q=z0/norm(z0); q2=q'; sigma=[];
relres=tol+1; iter=0;
```

```
while relres(end)>=tol && iter<=nmax
    iter=iter+1;
    b=P*q; y=L\b; z=U\y;
    q=z/norm(z); z=A*q; sigma=q'*z;
    b=q2'; y=U'\b; w=L'\y; q2=w'*P;
    q2=q2/norm(q2); costheta=abs(q2*q);
    if costheta>=5e-2
        temp=norm(z-sigma*q)/costheta;
        relres=[relres,temp];
    else
        fprintf('Valeur propre multiple'); break;
    end
    x=q;
end
return
```

Exemple: Méthode de la puissance inverse

```
clear; clc;
A = [15 -2 2; 1 10 -3; -2 1 0];
z0 = [1; 1; 1]./(3^0.5);
tol = 10^-12;
nmax = 1000;
mu = 1;
[sigma,x,iter,relres] = invpower(A,z0,mu,tol,nmax);
fprintf('sigma=%.15f\n',sigma);
fprintf('iter=%d\n',iter);
```

sigma=0.512084825571975
iter=10

4- LA MÉTHODE QR

Méthode QR

- Nous présentons dans cette section une méthode itérative pour approcher simultanément **toutes les valeurs propres** d'une matrice A donnée.
- L'idée de base est de transformer A en une matrice semblable pour laquelle le calcul des valeurs propres est plus simple.
- Le problème serait résolu si on savait déterminer de manière directe, c'est-à-dire en un nombre fini d'itérations, la matrice unitaire U de la décomposition de Schur, i.e. la matrice U telle que $\mathrm{T} = \mathrm{U} * \mathrm{AU}$, où T est triangulaire supérieure avec $t_{ii} = \lambda_i(\mathrm{A})$ pour $i = 1, \ldots, n$.
- Malheureusement, d'après le théorème d'Abel, dès que $n \geq 5$ la matrice U ne peut pas être calculée de façon directe.
- Notre problème ne peut donc être résolu que de manière **itérative**.

Méthode QR

- L'algorithme de référence sur ce thème est la **méthode QR** que nous n'examinerons que dans le cas des matrices réelles.
- Soit $A \in \mathrm{R}^{n\times n}$; on se donne une matrice orthogonale $\mathrm{Q}^{(0)} \in \mathrm{R}^{n\times n}$ et on pose $\mathrm{T}^{(0)} = (\mathrm{Q}^{(0)})^{\mathrm{T}}\mathrm{A}\mathrm{Q}^{(0)}$. Les itérations de la méthode QR s'écrivent pour $k = 1, 2, \ldots,$ jusqu'à convergence :

déterminer $\mathrm{Q}^{(k)}, \mathrm{R}^{(k)}$ telles que

$\mathrm{Q}^{(k)}\mathrm{R}^{(k)} = \mathrm{T}^{(k-1)}$ (factorisation QR);

puis poser

$\mathrm{T}^{(k)} = \mathrm{R}^{(k)}\mathrm{Q}^{(k)}$. **(5)**

- A chaque étape $k \geq 1$, la première phase consiste en la factorisation de la matrice $\mathrm{T}^{(\mathrm{k}-1)}$ sous la forme du produit d'une matrice orthogonale $\mathrm{Q}^{(\mathrm{k})}$ par une matrice triangulaire supérieure $\mathrm{R}^{(\mathrm{k})}$.

Méthode QR

déterminer $\mathrm{Q}^{(k)}, \mathrm{R}^{(k)}$ telles que

$\mathrm{Q}^{(k)}\mathrm{R}^{(k)} = \mathrm{T}^{(k-1)}$ (factorisation QR);

puis poser

$\mathrm{T}^{(k)} = \mathrm{R}^{(k)}\mathrm{Q}^{(k)}$.

(5)

La seconde phase est un simple produit matriciel. Remarquer que

$$\begin{aligned} \mathrm{T}^{(k)} &= \mathrm{R}^{(k)}\mathrm{Q}^{(k)} = {\mathrm{Q}^{(k)}}^{T}\left(\mathrm{Q}^{(k)}\mathrm{R}^{(k)}\right)\mathrm{Q}^{(k)} = {\mathrm{Q}^{(k)}}^{T}\mathrm{T}^{(k-1)}\mathrm{Q}^{(k)} \\ &= \left(\mathrm{Q}^{(0)}\mathrm{Q}^{(1)} \dots \mathrm{Q}^{(k)}\right)^{T}\mathrm{A}\left(\mathrm{Q}^{(0)}\mathrm{Q}^{(1)} \dots \mathrm{Q}^{(k)}\right), \qquad k \geq 0, \end{aligned} \quad \textbf{(6)}$$

autrement dit, toute matrice $\mathrm{T}^{(\mathrm{k})}$ est orthogonalement semblable à A. Ceci est particulièrement appréciable pour la stabilité de la méthode.

Méthode QR de base

Dans la forme “basique” de la méthode QR, on pose $\mathrm{Q}^{(0)} = \mathrm{I_n}$ de sorte que $\mathrm{T}^{(0)} = \mathrm{A}$. A chaque itération $k \geq 1$, la factorisation QR de la matrice $\mathrm{T}^{(\mathrm{k}-1)}$ peut être effectuée en utilisant le procédé de Gram-Schmidt modifié.

Exemple: On applique la méthode QR à la matrice symétrique $\mathrm{A} \in \mathrm{R}^{4\times4}$ telle que $a_{ii} = 4$, pour $i = 1, \ldots, 4$, et $a_{ij} = 4 + i - j$ pour $i < j \leq 4$, dont les valeurs propres sont $\lambda_1 = 11.09$, $\lambda_2 = 3.41$, $\lambda_3 = 0.9$ et $\lambda_4 = 0.51$. Après 20 itérations, on obtient:

$$\mathrm{T}^{(20)} = \begin{bmatrix} \boxed{11.09} & 6.44 \cdot 10^{-10} & -3.62 \cdot 10^{-15} & 9.49 \cdot 10^{-15} \\ 6.47 \cdot 10^{-10} & \boxed{3.41} & 1.43 \cdot 10^{-11} & 4.60 \cdot 10^{-16} \\ 1.74 \cdot 10^{-21} & 1.43 \cdot 10^{-11} & \boxed{0.9} & 1.16 \cdot 10^{-4} \\ 2.32 \cdot 10^{-25} & 2.68 \cdot 10^{-15} & 1.16 \cdot 10^{-4} & \boxed{0.58} \end{bmatrix}.$$

Méthode QR de base

```
function [T,Q,R]=basicqr(A, nmax)
% BASICQR Méthode QR de base
% [T,Q,R] = BASICQR(A,NMAX) effectue NMAX itérations de
% la version de base de la méthode QR
T=A;
for i=1:nmax
  [Q,R] = modgrams(T);
  T=R*Q;
end
return
```

Exemple: Méthode QR de base

```
clear; clc;
A = zeros(4,4);
 for i=1:4,for j=1:4
        if (i==j)
            A(i, j) = 4;
        else
            if (i<j)
                A(i, j) = 4+i-j; A(j, i) = 4+i-j;
            end
        end
End, end
 nmax = 30;
[T,Q,R]=basicqr(A, nmax)
```

```
T =
   11.0990    0.0000   -0.0000    0.0000
    0.0000    3.4142   -0.0000   -0.0000
    0.0000    0.0000    0.9010    0.0000
    0.0000    0.0000    0.0000    0.5858
Q =
    1.0000   -0.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000   -0.0000
    0.0000    0.0000    1.0000   -0.0000
    0.0000    0.0000    0.0000    1.0000
R =
   11.0990    0.0000   -0.0000    0.0000
         0    3.4142    0.0000   -0.0000
         0         0    0.9010    0.0000
         0         0         0    0.5858
```