Sorbonne Université

Rapport de projet

Calcul de Pseudospectre

Yang Zitong
Zheng Alix

Encadré par Stef Graillat

Table des matières

Introduction

Dans le cadre du projet "Calcul de pseudospectre", nous cherchons à tracer avec differents méthodes le pseudospectre d'une matrice carrée de taille quelconque. Nous allons ainsi définir ce qu'est le pseudospectre, puis présenter plusieurs méthodes pour les tracer et enfin rechercher l'abscisse pseudospectrale.

1 Qu'est-ce que le ε-pseudospectre d'une matrice ?

Le ε-pseudospectre d'une matrice $A \in \mathbb{C}^{n\times n}$, noté $\Lambda_\varepsilon(A)$, est défini comme le sous-ensemble du plan complexe contenant toutes les valeurs propres de toutes les matrices situées à une distance ε de A :

$$\Lambda_\varepsilon(A) =$$
$$\{z \in \mathbb{C} : z \in \Lambda(B) \quad avec \quad B \in \mathbb{C}^{n\times n} \quad et \ \|(B-A)\|_2 < \varepsilon\}$$

$\Lambda(B)$ est le spectre de B.

Pour toute matrice, M, à coefficient dans $\mathbb{C}$, il existe une factorisation de cette matrice de la forme :

$$M = U\Sigma V^*$$

avec U : une matrice unitaire, matrice de passage formée de vecteur propre de M
Σ : une matrice diagonale
V^* : matrice adjointe de V (matrice unitaire) -V^* est la matrice V conjuguée et transposée - la décomposition en valeurs singulières de M.

Définition :

Soit $M \in \mathbb{C}^{n\times n}$, $\|M\|_2 = max_{x\in\mathbb{C}^n} \frac{|Mx\|_2}{\|x\|_2}$

De plus, $\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : \sigma_{min}(A - z \cdot id) < \varepsilon\}$, qui permet de localiser le pseudospectre de la matrice A.
σ_{min} représente la plus petite valeur singulière. Nous appelons "valeur singulière", toutes les coefficients de la matrice $A - z.id$

2 Comment tracer le pseudospectre ?

2.1 Implantation de l'algorithme GRID

Dans un premier temps, nous cherchons à localiser les valeurs propres d'une matrice A. Pour localiser toutes les valeurs propres de la matrice, nous utilisons le théorème de Gerschgorin qui permet de borner les valeurs propres d'une matrice carrée.

Théorème : toute valeur propre de A appartient à au moins l'un des disques de Gerschgorin D_i défini comme suit :

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} a_{i,j} \right\}$$

Nous pouvons borner toutes les valeurs propres d'une matrice dans un plan complexe.
Puis, nous cherchons à localiser le pseudospectre. Pour cela, à partir de la formule de Gerschgorin, les coefficients de la matrice de départ la taille de la matrice et ε, nous avons borné les pseudo valeurs propres comme suit :

$$|\lambda - a_{ii}| \leq \sum_{i \neq j \ et \ i,j=0}^{n} |a_{ij}| + \varepsilon \cdot n$$

Démontrons la formule ci-dessus :

Soit $\lambda \in \mathbb{C}$, $\quad \sigma_\varepsilon = \{z \in \mathbb{C} : \|(z \cdot id - A)^{-1}\| > \epsilon^{-1}\}$ $A \in M_n(\mathbb{C})$
Si $\lambda \in \sigma_\varepsilon(A)$ alors il existe $B \in M_n(\mathbb{C})$ telle que : $\|B - A\| \leq \varepsilon$
et λ est une valeur propre de la matrice B
Le disque de Gerschgorin : $D_i = \left\{ z \in \mathbb{C} \ : |z - a_{ii}| \leq \sum_{j \neq i} a_{i,j} \right\}$

Encadrons $|\lambda - a_{ii}|$ en fonction des coefficients de la matrice A, taille de A et de la perturbation ε :

$|\lambda - a_{ii}| = |\lambda - a_{ii} + b_{ii} - b_{ii}| \leq |\lambda - b_{ii}| + |b_{ii} - a_{ii}|$. (inégalité triangulaire)
Nous savons que $|\lambda - b_{i,i}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |b_{i,j}|$,
or, $\forall i, j \ |b_{ij} - a_{ij}| \leq \sup(b_{ij} - a_{ij}))$,
donc $\sum_{i,j=0}^{n} |b_{ij} - a_{ij}| \leq n \cdot sup(b_{ij} - a_{ij})$ et $|b_{ij} - a_{ij}| \leq |b_{ij}| - |a_{ij}|$,
d'où $|b_{ij}| \leq |b_{ij} - a_{ij}| + |a_{ij}|$.

Ainsi nous obtenons :

$$\sum_{i,j=0}^{n} |b_{ij}| \leq \sum_{i,j=0}^{n} |b_{ij} - a_{ij}| + \sum_{i,j=0}^{n} |a_{ij}| \ , \leq \sum_{i,j=0}^{n} |a_{ij}| + (n-1) \cdot (sup(b_{ij} - a_{ij}))$$

mais nous savons que

$$sup(b_{ij} - a_{ij}) \leq |B - A| \leq \varepsilon \text{ donc } \sum_{i,j=0}^{n} |b_{ij}| \leq \sum_{i,j=0}^{n} |a_{ij}| \ + n \cdot (\varepsilon) \ .$$

De plus, nous avons $|b_{ij} - a_{ij}| \leq sup(b_{ij} - a_{ij}) \leq \varepsilon$,
donc nous obtenons que $|\lambda - a_{ii}| \leq \sum_{i,j=0}^{n} |a_{ij}| \ + (n-1) \cdot \varepsilon + \varepsilon$,

ou encore

$$|\lambda - a_{ii}| \leq \sum_{i \neq j \ et \ i,j=0}^{n} |a_{ij}| \ + \varepsilon \cdot n \ .$$

Puis, nous cherchons à trouver une grille contenant toutes ses pseudo valeurs propres. Pour cela, nous comparons toutes les parties réelles et les parties imaginaires de chaque borne de pseudo valeur propre. Ainsi, nous pouvons trouver la borne superieure et inferieure de la grille.

Enfin, pour chaque point z de la grille, nous calculons la valeur minimale des valeurs singulières de la matrice $z * I - A$ et nous utilisons contour pour tracer le pseudospectre.

• Voici le pseudocode pour obtenir la grille contenant toutes les valeurs singulières :

local_vp
Entrées : $A = (a_{ij}) \in \mathbb{C}^{n \times n}$, taille
Sortie : [xsup,xinf,ysup,yinf]
Corps :

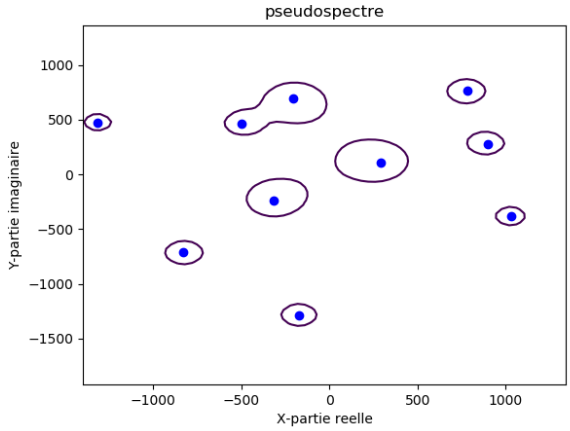
pour $i < taille$ **faire** : ray=0
pour $j < taille\ j \neq i$ **faire** : $ray+ = |a_{ij}|$
si $i = 0$ **alors** :
$xsup \leftarrow a_{00}.real + ray \quad xinf \leftarrow a_{00}.real - ray$
$ysup \leftarrow a_{00}.imag + ray \quad yinf \leftarrow a_{00}.imag - ray$
sinon :
si $a_{ii}.real + ray > xsup$ **alors** :$xsup \leftarrow a_{ii}.real + ray$
si $a_{ii}.real - ray < xinf$ **alors** :$xinf \leftarrow a_{ii}.real - ray$
si $a_{ii}.imag + ray > ysup$ **alors** : $ysup \leftarrow a_{ii}.imag + ray$
si $a_{ii}.imag - ray < yinf$ **alors** :$yinf \leftarrow a_{ii}.imag - ray$

Les valeurs des bornes de la grille sont initialisées au premier coefficient diagonal de la matrice qui correspond au centre du cercle de Gershgorin,
ray correspond au rayon du cercle de Gershgorin,
a_{ii} correspond à la valeur du coefficient de la diagonale de la matrice A.
Pour obtenir la grille contenant le pseudospectre, il suffit d'ajouter ou enlever $\varepsilon.n$ à la borne de la grille.

- Voici le pseudocode pour obtenir la grille contenant le pseudospectre :

locaPseudoVp
Entrées : A $\in \mathbb{C}^{n\times n}$, e (ε) > 0 ,taille
Sortie : $[xsup + e \times taille, xinf - e \times taille, ysup + e \times taille, yinf - e \times taille]$
Corps : loca_vp(A,taille)

Image d'un pseudospectre d'une matrice quelconque



2.2 Optimisation de l'algorithme GRID

Après avoir implémenté l'algorithme GRID qui consiste à parcourir la grille contenant toutes les pseudo-valeurs propres et à tracer le contour de chaque point vérifiant l'équation : $\sigma_{min}(A - z \cdot id) = \varepsilon$.
Nous cherchons à optimiser le parcours de la grille, lorsque nous savons que certains pseudo-valeurs propres sont connexes. Nous pouvons trouver plusieurs sous-grilles contenant plusieurs pseudo-valeurs propres et ainsi éviter des parcours de grilles inutiles.

Pour cela, nous créons des listes de bornes, c'est-à-dire pour chaque disque de Gerschgorin, nous regardons s'il coïncide avec les disques déjà calculé. Lorsqu'un disque coïncide avec un autre disque, nous les regroupons en stockant les bornes supérieures et inférieures pour x et y . Pour savoir si deux disques sont connexes, nous comparons les distances entre le centre de chaque disque.

• Voici le pseudo-code pour localiser le pseudospectre d'une matrice carrée A de façon optimisée :

localOPTI

Entrées : $A \in \mathbb{C}^{n\times n}$, $\varepsilon > 0$, taille
Sortie : newborne = liste des composants des bornes
ie : [[xsup, xinf, ysup, yinf], ...]
Corps :
initialisation :
Ldisque← liste des cercles Gerschgorins :
[[a_{ii} (centre) , $\sum_{j\neq i} a_{ij} + \varepsilon \times taille(delamatrice)$], ...]

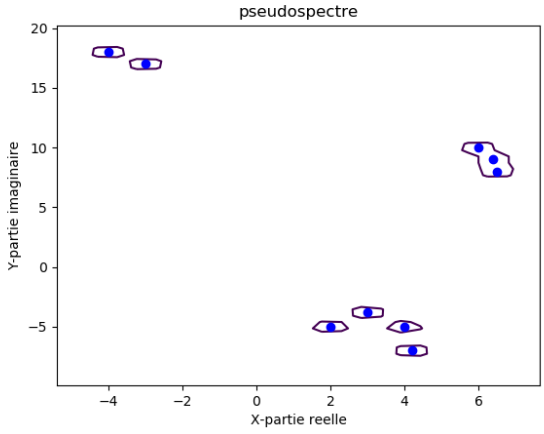
Étape 1 : chercher les ensembles qui contiennent des disques liés directement
d=[ ] : liste des ensembles des disques en relation directe
pour d_1 dans Ldisque **faire** :
b= $\{(d_1[0], d_1[1])\}$: ensembles des disques liés avec d_1
pour d_2 dans Ldisque $d_2 \neq d_1$ **faire** :
si $|d_1[0] - d_2[0]| < d_1[1] + d_2[1]$ **alors** :
ajouter $(d_2[0], d2[1])$ dans l'ensemble b
si b n'est pas dans d **alors** :
ajouter b dans d

Étape 2 : vérifier les relations indirectes que peuvent avoir les ensembles de d
pour d_1 dans d **faire** :
pour d_2 dans d $d_2 \neq d_1$ **faire** :
si $d_1 \cap d_2 \neq \emptyset$ **alors** :

$d_1 \leftarrow d_1 \cup d_2$
supprimer d_2 dans d

Étape 3 : calculer la borne pour chaque composant
borne=[ ] : [[xsup,xinf,ysup,yinf], ...]
pour d_1 dans d **faire** :
calculer [xsup,xinf,ysup, yinf] pour chaque composant (d_1) et l'ajouter dans borne
Étape 4 : vérifier de nouveau si les bornes sont adjoints ou non
pour i dans borne **faire** :
pour j dans borne $j \neq i$ **faire** :
si |(i[0]+i[1])/2-(j[0]+j[1])/2| <(i[0]-i[1])/2+(j[0]-j[1])/2 **alors** :
si|(i[2]+i[3])/2-(j[2]+j[3])/2| <(i[2]-i[3])/2+(j[2]-j[3])/2
alors :
$i[0] \leftarrow max(i[0], j[0])$ $i[1] \leftarrow min(i[1], j[1])$
$i[2] \leftarrow max(i[2], j[2])$ $i[3] \leftarrow min(i[3], j[3])$
newborne=[ ]
pour i dans borne **faire** :
si i n'est pas dans newborne **alors** l'ajouter dans newborne

Image du pseudospectre d'une matrice complexe pouvant appliquer la méthode d'optimisation de l'algorithme de GRID



2.3 Méthode de Prédiction-Correction

À partir des valeurs propres d'une matrice M que nous souhaitons tracer le pseudo-spectre, nous allons pour chaque valeur propre de la matrice :

1. trouver le premier point du contour vérifiant $g(z) = \varepsilon$
2. prédire les autres points z_k
3. corriger les points

Étape 1 Pour trouver le premier point z_0, il faut :

-choisir une direction d_0, le coefficient directeur (d_0) doit être de norme 1

-prendre le point qui intersecte la droite qui a pour coefficient directeur d_0 et la courbe $g(z) = \varepsilon$

Étape 2 Pour prédire le point suivant, $\tilde{z}_k$, il faut :

-choisir une direction arbitraire à partir du point précédent, z_{k-1}

-prendre la droite tangente r_k qui est égale à $\frac{i\bigtriangledown g(\tilde{z}_{k-1})}{|g(\tilde{z}_{k-1})|} = \frac{iv^*_{min}u_{min}}{v^*_{min}u_{min}}$

-$z_k = z_{k-1} + r_k\tau_k$, τ_k est la longueur séparant deux points

Étape 3 Pour corriger le point $\tilde{z}_k$, il faut :

-utiliser la méthode de Newton, nous appliquons la formule :

$$z_k = \lambda - \frac{\sigma_{min} - \varepsilon}{u^*_{min}v_{min}}$$

z_k est le point corrigé, $\sigma_{min} = g(\lambda)$
le gradient : $\bigtriangledown g(\lambda) = v^*_{min}u_{min}$
$g(\lambda) = \sigma_{min}(zI - M)$

- Voici le pseudocode pour la méthode de Prédiction-Correction :

Prédiction-Correction

Entrées : $A \in \mathbb{C}^{n\times n}$, $\varepsilon > 0$
Sortie : z : liste de tous les points z_k sur $\partial \wedge_\varepsilon (A)$
Corps :

initialisation : $tol << 1$

$d_0 \leftarrow 1 + 0 \times i$ (la direction pour le premier point)

theta0(θ_0)$\leftarrow \varepsilon$

pour chaque eigenvalue(eig(λ_0)) de A **faire** :

z1new (z_1^{new}) $\leftarrow eig(\lambda_0) + \theta_0 d_0$

tant que $|g(z_1^{new}) - \varepsilon|/\varepsilon > tol$:

$z_1^{old} \leftarrow z_1^{new}$

calculer svd de $z_1^{old} I - A$ et prendre $(\sigma_{min}, u_{min}, v_{min})$

$z_1 \leftarrow z_1^{new}$ et $z_k \leftarrow z_1^{new}$

tant que $|z_k - z_1^{new}|/t > t_k$ **faire** :

$r_k = iv_{min}^* u_{min}/|v_{min}^* u_{min}|$

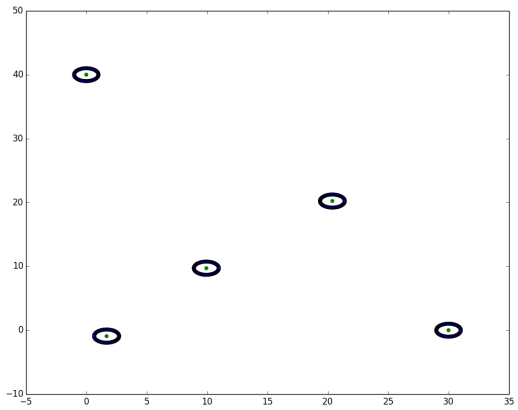
$t_k = min(0.1, \frac{1}{2} dist(z_k, \wedge(A)))$

$z_1(\widetilde{z_k}) \leftarrow z_k + t_k r_k$

calculer svd de $z_1 I - A$ et prendre $(\sigma_{min}, u_{min}, v_{min})$

$z_k \leftarrow \lambda_0 - \frac{\sigma_{min} - \varepsilon}{u_{min}^* v_{min})} d_0$

Image du pseudospectre par la méthode de prediction-correction



2.4 Méthode du pseudospectre par composante

À partir de la grille contenant toutes les valeurs propres, nous implémentons l'ensemble :

$$\{\lambda \in \mathbb{C} : \rho(\,|(A-\lambda I)^{-1}|\,E\,) > \varepsilon^{-1}\} \quad (1)$$

Avec $A \in M_n(\mathbb{C})$
$E = |A|$

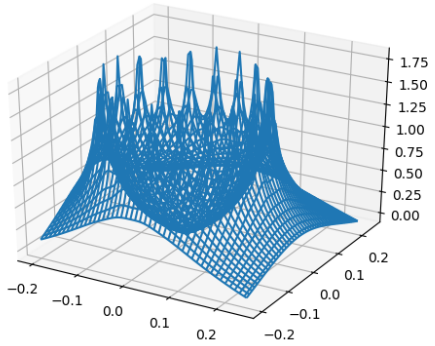
Définition :

Soit $M \in \mathbb{C}^{n\times n}$, $M = (m_{ij})$, $|M| = (|m_{ij}|)$

$\rho(A) = max_{0\leq i\leq n}(|\lambda_i|)$, c'est le rayon spectrale

En effet, pour chaque point de la grille, nous trouvons le rayon spectral pour la matrice $(A-\lambda * I)^{-1}$ et nous utilisons contour pour tracer le pseudospectre.

Image du pseudospectre d'une matrice 20×20, où $a_{i,j+1} = -1$ *pour i=1 :19 et* $a_{20,1} = 2^{-56}$



3 Recherche de l'abscisse pseudospectrale

Nous recherchons le point avec la plus grande partie réelle sur le pseudospectre d'une matrice A de taille $n \in \mathbb{N}$. l'abscisse pseudospectrale $\alpha_\varepsilon(A)$ est definie par :

$$\alpha_\varepsilon(A) = sup_{z \in \sigma_\varepsilon(A)}\ Re(z)$$

Pour le faire déterminer l'abscisse pseudospectrale, nous allons :

1. trouver la valeur propre λ la plus à droite
2. trouver le point qui intersecte la droite $Im(\lambda)$ avec le contour de λ

   Pour tout k entier $k \in$ 1,2,...
3. trouver tout les points qui intersecte la droite verticale, $Re(z_k)$ et ainsi déterminer les intervalles
4. trouver le point le plus à droite de chaque intervalle

Étape 1 Pour trouver le point le plus à droite des valeurs propres, il suffit de prendre le point qui a la plus grande partie réelle (ie : $max(Re(\lambda)) = \alpha(A)$)

Étape 2 Pour trouver le point qui intersecte la droite verticale il faut :

-choisir la direction $d_0 = 1$,

-prendre le point qui intersecte la droite qui a pour coefficient directeur d_0 et la courbe $g(z) = \varepsilon$

Étape 3 Pour trouver les points de la forme $Re(z_k) + iy_i$ qui intersecte la droite verticale, $Re(z_k)$, il faut :

-chercher les y_i

Pour cela, il suffit de trouver les valeurs propres de la matrice de taille $2 \times n$:

$$\begin{pmatrix} Re(z_k) - A^* & \text{-}\varepsilon I \\ \varepsilon I & A - Re(z_k) \end{pmatrix}$$

dont la partie réelle est nulle.

Les y_i sont la partie imaginaire des valeurs propres trouvées.

-trier les points par ordre croissant de la partie imaginaire :

Nous obtenons ainsi une liste de m valeurs, avec $m < 2 \times n$:

$$[y_1, y_2, \ldots, y_m]$$

Les intervalles sont constitués ainsi :

$$[y_1,\ y_2]$$
$$\ldots$$
$$[y_{m-1},\ y_m]$$

Certains intervalles sont constitués que d'un point.

Étape 4 Pour trouver le point le plus à droite, il faut :

-prendre le point qui correspond au milieu de chaque intervalle

-à partir de ce point, chercher le point le plus à droite comme à l'étape 3.

- Voici le pseudocode pour trouver l'abscisse pseudospectrale :

fonction : $\widehat{Y}$ $(((x+iy)-A))$
Entrées : $x \in \mathbb{C}$, $A \in \mathbb{C}^{n\times n}$, taille , $\varepsilon > 0$
Sortie : renvoyer la liste $(\widehat{Y})$ contenant tous les points $\hat{y}_j$ tel que :
i$\hat{y}_j$ est un valeur propre purement imaginaire pour la matrice H :

$$\begin{pmatrix} x - A^* & -\varepsilon I \\ \varepsilon I & A - x \end{pmatrix}$$

Corps : construire la matrice H et calculer les valeurs propres (λ_H)
pour chaque (λ_{iH}) **faire** :
si Re λ_{iH}=0 **alors** : ajouter dans la liste $\widehat{Y}$

abscisse :
Entrées : $A \in \mathbb{C}^{n\times n}$, $\varepsilon > 0$
Sortie : Rez_k
$\alpha_\varepsilon(A) = supRe(z),\ z \in \sigma_\varepsilon(A)$

Corps :

Étape 1 : trouver le valeur propre de A la plus à droite (λ)

$\alpha(A) \leftarrow supRe(\lambda_i)$, $\lambda_i \in \sigma(A)$

pour chaque valeur propre de $A(\lambda_i)$ **faire** :

si $Re\lambda_i = \alpha(A)$ **alors** : $\lambda = \lambda_i$

Étape 2 :

Initialisation : $d_0 \leftarrow 1 + 0 \times i$

tol$\leftarrow e^{-14}$

$(\theta_0) \leftarrow \varepsilon$

z1new$(z_1^{new}) \leftarrow \lambda + \theta_0 d_0$

tant que $|g(z_1^{new}) - \varepsilon| > tol * \varepsilon$ **Faire** :

$z_1^{old} \leftarrow z_1^{new}$

calculer la décomposition singulière de $z_1^{old} I - A$ et prendre

$(\sigma_{min}, u_{min}, v_{min})$

$z_1^{new} \leftarrow \lambda_0 - \dfrac{\sigma_{min} - \varepsilon}{Re(\bar{d} u_{min}^* v_{min})} d_0$

$z_1 \leftarrow z_1^{new}$

tant que ($y_{MAX} - y_{MIN} > tol$) **Faire** :

Étape 3 et 4 : $\hat{y} \leftarrow \widehat{Y}$((Rezk+iy)-A)

pour $\hat{y}_j$ dans $\hat{y}$ **faire** :

si $\sigma_{min}(\hat{y}_j) = \varepsilon$ **alors** : ajouter dans la liste y

trier y ($y_1 ... < y_j ... < y_n$)

$interval \leftarrow [[y_1, y_2], [y_j, y_{j+1}] ... [y_{n-1}, y_n]]$

$midpoint \leftarrow [(y_1 + y_2)/2, (y_3 + y_4)/2 ... (y_j + y_{j+1})/2]$

pour m_j dans midpoint **faire** :

$\hat{z}_j \leftarrow max(Y(i(x + i \times m_j - A))$

zk$\leftarrow \hat{z}_0$

pour j = 1...n **faire** : si $Rez_k < Re\hat{z}_j$ alors $z_k = \hat{z}_j$

Conclusion

Pour conclure, nous avons réussi à connaitre et à mieux comprendre la notion de pseudospectre à travers ce projet. Nous avons étudié et tracé de plusieurs façon le pseudospectre de matrices complexes quelconque. L'algorithme de prédiction et correction étant le plus performant par rapport à l'algorithme de GRID en terme de rapidité.

Remerciement

Nous tenons à remercier notre encadrant, Stef Graillat qui nous a aidées et qui a répondu à nos questions tout au long du semestre.

Bibliographie

Définitions et théorèmes

[1] https ://fr.wikipedia.org/wiki/Th
[2] http ://citron.9grid.fr/docs/gerschgorin.pdf

Programmation

[1] http ://mathesaurus.sourceforge.net/matlab-numpy.html
[2] http ://www.ac-grenoble.fr/disciplines/maths/pages/PM/Ressources

/300/Creer_une_matrice_en_Python.pdf
[3] http ://math.mad.free.fr/depot/numpy/base.html
[4] https ://www.courspython.com/tableaux-numpy.htmlla-fonction-numpy-size
[5] https ://fr.mathworks.com/help/matlab/ref/contour.html
[6] https ://stackoverflow.com/questions/7375858/how-do-i-generate-contour-plot-from-a-n-x-n-matrix
[7] https ://fr.mathworks.com/help/matlab/ref/contour.html
[8] http ://www.runoob.com/python/python-tutorial.html

Documents fournis

[1] Martin Brühl. A curve tracing algorithm for computing the pseudospectrum. BIT, 36(3) :441– 454, 1996.
[2] A. N. Malyshev and M. Sadkane. Componentwise pseudospectrum of a matrix. Linear Algebra Appl., 378 :283–288, 2004.
[3] Lloyd N. Trefethen and Mark Embree. Spectra and pseudospectra. Princeton University Press, Princeton, NJ, 2005.