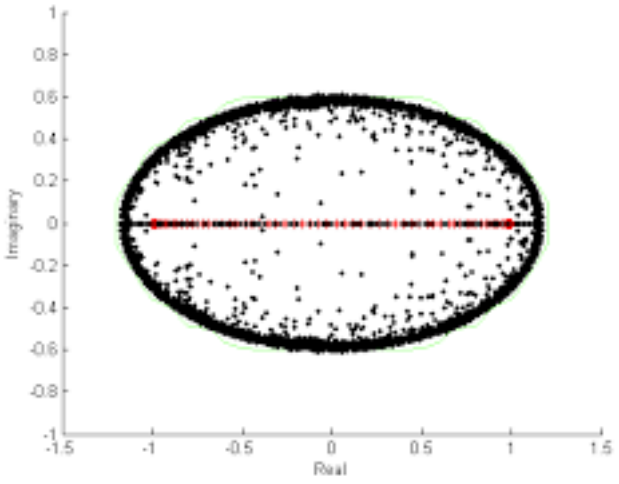
SORBONNE UNIVERSITÉ

POLYTECH SORBONNE

Sorbonne Université - Polytech-Sorbonne

Rapport Projet d'inititation : Pseudo spectre

Astrid Legay, Garance Morand, Ibtissam Lachhab
MAIN3
Encadrant : Stef Graillat



Année 2018-2019

Nous tenons à remercier notre tuteur Mr Stef Graillat pour son aide et sa participation.
Date de rendu, Mai 2019

Table des matières

1 Introduction

Dans le cadre de notre première année d'école d'ingénieur à Polytech Sorbonne, dans la filière mathématiques appliquées et informatique numérique, nous avons des projets pluridisciplinaires à faire. Ces derniers consistent à unir plusieurs matières pour réaliser un projet donné et supervisé par un tuteur.
Notre projet supervisé par Monsieur Graillat, avait pour but de déterminer à l'aide d'algorithmes le pseudo-spectre d'une matrice donnée.
Depuis une dizaine d'années, il y a eu dans le domaine des mathématiques numériques un vif intérêt porté à l'étude de la notion de pseudo-spectre. Le développement de cette notion s'explique par le fait que dans un certain nombre de problèmes d'ingénierie mathématique on note de sensibles différences entre d'un côté les résultats théoriques et les prédictions suggérées par l'analyse spectrale de ces opérateurs, et d'un autre côté les résultats obtenus par simulation numérique. Ce constat originel laisse penser que dans certains cas, la connaissance seule du spectre d'un opérateur ne permet pas de comprendre suffisamment son action. C'est ainsi que pour suppléer à cet apparent manque d'information contenu dans le spectre, de nouveaux sous-ensembles du plan complexe appelés pseudo-spectres ont été introduits.
L'étude des valeurs propres et des pseudo valeurs propres de matrices sert dans de nombreux domaines comme la résolution d'équations différentielles ou la mécanique quantique. Le calcul du pseudo-spectre, c'est-à-dire de l'ensemble des pseudo valeurs propres permet de compléter les informations données par le spectre. Etudier le comportement du pseudo-spectre d'une matrice revient à étudier la différence de comportement en pratique et en théorie des valeurs propres d'une matrice. Il existe de nombreux algorithmes permettant de calculer le pseudo-spectre d'une matrice. Nous en avons réalisé deux, que nous allons présenter ici, le premier basé sur les disques de Gerschgorin, et le second sur la méthode de prédiction-correction de Newton.

2 Rappel sur les valeurs propres et notions des pseudo-valeurs propres

Nous allons ici vous rappeler certaines notions comme les valeurs propres, le spectre d'une matrice ou autre. Puis nous définirons de nouvelle notions concernant les pseudo valeurs propres et les pseudo-spectres. Cette partie constitue un outil essentiel pour comprendre la suite du rapport.

2.1 Notions de valeur propre et de vecteur propre

Définition

Soit $A \in M_n(\mathbb{C})$ et $\lambda \in \mathbb{C}$. On dit que λ est valeur propre de A s'il existe un vecteur $x \in \mathbb{C}^n, x \neq 0_{\mathbb{K}^n}$, tel que $Ax = \lambda x$. Le vecteur x est appelé vecteur propre de A associé à la valeur propre λ.

Définition

On appelle spectre d'une matrice $A \in M_n(\mathbb{C})$ l'ensemble de ses valeurs propres de A dans $\mathbb{C}$ considéré. On le note $Spec(A)$.

$$\text{Comme } Ax = \lambda x \iff (A - \lambda I_n)x = 0_{\mathbb{C}^n} \iff x \in Ker(A - \lambda I_n),$$

on en déduit que x est un vecteur propre de A associé à la valeur propre λ si x appartient à l'ensemble

$$V_\lambda := Ker(A - \lambda I_n)$$

et que λ est une valeur propre de A si il existe $x \neq 0_{C^n}$ appartenant à V_λ, c'est-à-dire si $V_\lambda \neq \{0_{C^n}\}$.

Définition

Le sous-espace vectoriel V_λ est appelé sous-espace propre de A associé à la valeur propre λ.

Proposition

Une matrice $A \in M_n(\mathbb{C})$ possède au plus n valeurs propres distinctes.

2.2 Matrices diagonalisables

Définition (Matrice diagonalisable)

Une matrice $A \in M_n(\mathbb{C})$ est dite diagonalisable si elle est semblable à une matrice diagonale, c'est-à-dire s'il existe une matrice $P \in M_n(\mathbb{C})$ inversible telle que

$$D = P^{-1}AP$$

soit une matrice diagonale. Ainsi une matrice est diagonalisable si l'on peut la factoriser sous la forme :

$$A = PDP^{-1}$$

où D est une matrice diagonale, dont les coefficients diagonaux correspondent aux valeurs propres de la matrice A. De plus, la matrice $P \in M_n(\mathbb{C})$ est inversible et correspond à la matrice de passage de la base canonique à la base formée par les vecteurs propres de A. Ainsi, on peut également définir la diagonalisabilité d'une matrice de la sorte : Une matrice $A \in M_n(\mathbb{C})$ est diagonalisable s'il existe une base de $\mathbb{C}^n$ formée de vecteurs propres de A.

2.3 Décomposition en valeurs singulières

Théorème

Soit $A \in M_{m,n}(\mathbb{C})$, alors il existe $U \in M_m(\mathbb{C})$ et $V \in M_n(\mathbb{C})$ deux matrices orthogonales ($i.e., U^* = U^{-1}$ et $V^* = V^{-1}$, et une matrice $\Sigma \in M_{m,n}(\mathbb{C})$ diagonale ($i.e., \Sigma_{i,j} = 0$ si $i \neq j$ et $\Sigma_{i,i} = \sigma_i$ avec

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{min(m,n)} \geq 0,$$

telles que

$$A = U\Sigma V^*,$$

Les valeurs $\sigma_1, \ldots, \sigma_{min(m,n)}$ sont appelées valeurs singulières de A. Les $min(m,n)$ dernières colonnes $(u_i)_{1 \leq i \leq min(m,n)}$ et $(v_i)_{min(m,n)}$ de U et V sont appelés vecteurs singuliers associés aux valeurs singulières à gauche et droite respectivement.

La matrice V contient un ensemble de vecteurs de base orthonormée de $\mathbb{K}^{\times}$, dits « d'entrée » ou « d'analyse ». La matrice U contient un ensemble de vecteurs de base orthonormée de $\mathbb{K}^{\triangleright}$, dits « de sortie ». La matrice Σ contient dans ses coefficients diagonaux les valeurs singulières de la matrice M.

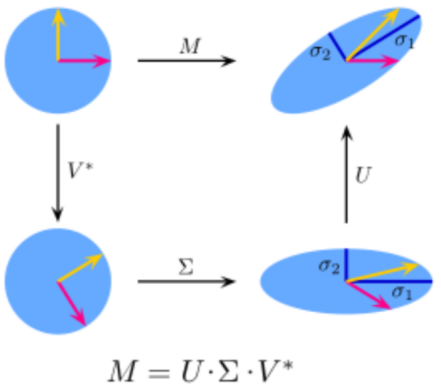


FIGURE 1 – Schéma pour expliquer la méthode de décomposition en valeurs singulières

Pour ce qui est du calcul de SVD, le calcul explicite, analytique, de la décomposition en valeurs singulières d'une matrice est difficile dans le cas général. On utilise, en particulier dans les applications, des algorithmes spécialisés. Si la matrice possède plus de lignes que de colonnes, on effectue tout d'abord une décomposition QR. Le facteur R est ensuite réduit sous forme bidiagonale. Pour ceci, on peut effectuer des transformations de Householder alternativement sur les colonnes et sur les lignes de la matrice. Les valeurs singulières et vecteurs singuliers sont alors trouvées en effectuant une itération de type QR bidiagonale.

2.4 Pseudo valeurs propres et pseudo-spectre

Nous allons maintenant définir rapidement ce que sont les pseudo valeurs propres et le pseudo spectre d'une matrice afin d'obtenir des conditions sur les pseudo spectres qui nous permettront d'implémenter nos algorithmes.

Première définition

Considérons A une matrice appartenant à $\mathbb{C}^{n\times n}$ et $\varepsilon > 0$ quelconque. Le pseudo-spectre $\sigma_\varepsilon(A)$ de A est l'ensemble des $z \in \mathbb{C}$ tel que :

$$||(zI - A)^{-1}|| > \varepsilon^{-1}$$

Deuxième définition

$\sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que :
$z \in \sigma(A + E)$ avec E telle que $E \in \mathbb{C}^{N\times N}$ et $||E|| < \varepsilon$.

Troisième définition

$\sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que : $||(z - A)v|| < \varepsilon$ pour un $v \in \mathbb{C}^N$ quelconque avec $||v|| = 1$.

Théorème

Pour n'importe quelle matrice A appartenant à $\mathbb{C}^{n\times n}$, les trois premières définitions sont équivalentes.

Quatrième définition du pseudo-spectre

Pour $||.|| = ||.||_2, \sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que

$$\sigma_{min}(z - A) < \varepsilon.$$

3 Premier algorithme

3.1 Théorème de Gerschgorin

Le théorème de Gerschgorin stipule que toute valeur propre d'une matrice appartient à au moins l'un des disques de Gerschgorin. Nous allons nous servir des définitions vues précédemment afin de démontrer que ce théorème s'applique également pour les pseudo valeurs propres.

Enoncé

Considérons une matrice complexe de taille n^2 et de terme général (a_{ij}). Pour chaque indice de ligne i entre 1 et n on introduit le disque de Gerschgorin correspondant :

$$D_i = z \in \mathbb{C}, |a_{ii} - z| \leq \sum_{j \neq i}(|a_{ij}|) = D(a_{ii}, R_{ii}).$$

Théorème

Toute valeur propre de A appartient à au moins l'un des disques de Gerschgorin.

Appliqué au pseudo-spectre

Première définition

Considérons A une matrice appartenant à $\mathbb{C}^{n \times n}$ et $\varepsilon > 0$ quelconque. Le pseudo-spectre $\sigma_\varepsilon(A)$ de A est l'ensemble des $z \in \mathbb{C}$ tel que :

$$||(zI - A)^{-1}|| > \varepsilon^{-1}.$$

D'après la définition, le pseudo-spectre est donc le sous-ensemble ouvert du plan complexe délimité par la courbe de niveau ε de la norme de la résolvante qui est la matrice $(z - A)^{-1}$ Ainsi donc, son étude se réduira à l'étude des lignes de niveau de la norme de sa résolvante.

Deuxième définition

$\sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que :
$z \in \sigma(A + E)$ avec E telle que $E \in \mathbb{C}^{N \times N}$ et $||E|| < \varepsilon$.

D'après cette définition, le pseudo-spectre est donc l'ensemble des nombres qui sont les valeurs propres d'une matrice qui a subi des perturbations $A + E$ avec $||E|| < \varepsilon$.

Troisième définition

$\sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que : $||(z - A)v|| < \varepsilon$ pour un $v \in \mathbb{C}^N$
quelconque avec $||v|| = 1$

Enoncé

Soit λ pseudo valeur propre de $A = (a_{ij})$ de dimension $n \times n$, donc par définition : $\lambda \in \sigma_\varepsilon(A)$ avec $\sigma_\varepsilon(A)$ l'ensemble des pseudo valeurs propres de A, nommé le pseudo spectre de A. On a :

$$|\lambda - a_{ii}| \leq n * \varepsilon + \sum_{i \neq j}^{n} |a_{ij}|.$$

Démonstration

D'après la deuxième définition du pseudo spectre, $\sigma_\varepsilon(A)$ est l'ensemble des $z \in \mathbb{C}$ tel que :
$z \in \sigma(A + E)$ avec E telle que $E \in \mathbb{C}^{N \times N}$ et $||E|| < \varepsilon$. D'où en posant $E = B - A$, on a bien l'implication suivante :

$$\lambda \in \sigma_\varepsilon(A) \Leftrightarrow \lambda \text{ valeur propre de } B \text{ telle que } ||B - A|| < \varepsilon.$$

De plus, d'après le théorème de Gerschgorin qui énonce que toute valeur propre de A appartient à l'un au moins des disques de Gerschgorin, on peut en déduire que si λ est une valeur propre de B alors il existe un disque de Gerschgorin auquel elle appartient. Ce qui se traduit comme suit.

$$\lambda \text{ valeur propre de } B \Leftrightarrow \exists\ i \text{ tel que } |\lambda - b_{ii}| \leq \sum_{i \neq j}^{n} (|b_{ij}|).$$

Dès lors, en usant de l'astuce $\lambda - a = \lambda - b + b - a$ et en utilisant la propriété de l'inégalité triangulaire, on obtient donc :

$$|\lambda - a_{ii}| \leq |\lambda - b_{ii}| + |b_{ii} - a_{ii}|.$$

De plus, en réutilisant le théorème de Gerschgorin, on l'inégalité qui suit

$$|\lambda - a_{ii}| \leq \sum_{i=1, i \neq j}^{n} (|b_{ij}|) + |b_{ii} - a_{ii}|.$$

Ensuite, en utilisant de nouveau l'inégalité triangulaire pour la valeur absolue et la linéarité de la somme, on obtient :

$$|\lambda - a_{ii}| \leq \sum_{i \neq j}^{n} |b_{ij} - a_{ij}| + \sum_{i \neq j}^{n} |a_{ij}| + |b_{ii} - a_{ii}|$$

On cherche désormais à majorer le coefficient $|b_{ij} - a_{ij}|$ de la matrice $B - A$ Or, nous savons d'après les hypothèses faites sur les perturbations de la matrice A que $||E|| < \varepsilon$. Notre perturbation étant $E = B - A$, nous avons donc $||B - A|| < \varepsilon$ d'où en usant de la norme euclidienne matricielle pour expliciter notre terme, on obtient :

$$||B - A|| = sup_{x \in \mathbb{C}^\kappa, x \neq 0} \frac{||(B-A)x||}{||x||} < \varepsilon$$

Si la borne supérieure de $\frac{||(B-A)x||}{||x||}$ est inférieure à ε alors :

$$\forall x \neq 0 \in \mathbb{C}^n, \frac{||(B-A)x||}{||x||} < \varepsilon$$

Pour simplifier l'écriture, on pose $B - A = E$ Ainsi on pose $x = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$,

avec 1 se trouvant à la $j_{ème}$ ligne. Ainsi, le produit matriciel donne :

$$\frac{||Ex||}{||x||} = || \begin{pmatrix} e_{1j} \\ \vdots \\ e_{ij} \\ \vdots \\ e_{nj} \end{pmatrix} ||$$

et ainsi :

$$\frac{||Ex||}{||x||} = \sqrt{\sum_{k=1}^{n} e_{kj}^2}$$
$$\Leftrightarrow$$
$$\sqrt{\sum_{k=1}^{n} e_{kj}^2} < \varepsilon$$
$$\Leftrightarrow$$
$$\sum_{k=1}^{n} e_{kj}^2 < \varepsilon^2$$
$$\Leftrightarrow$$
$$\forall k \in [1, n], e_{kj}^2 < \varepsilon^2.$$

D'où en particulier avec $k = j$, on a : $|e_{ij}| < \varepsilon$.

On a donc montré que pour toute matrice E, $||.||$ la norme euclidienne et ε quelconque, on a :

$$||E|| < \varepsilon \Rightarrow |e_{ij}| < \varepsilon.$$

Ainsi, on a avec $E = B - A$:

$$||B - A|| < \varepsilon \rightarrow |b_{ij} - a_{ij}| < \varepsilon.$$

On reprend donc le cours de notre démonstration :

$$|\lambda - a_{ii}| \leq \sum_{i \neq j}^{n} (|b_{ij} - a_{ij}|) + \sum_{i \neq j}^{n} (|a_{ij}|) + |b_{ii} - a_{ii}|.$$

A l'aide du résultat de notre démonstration ci-avant, soit : $\forall i, j \in [1, n], |b_{ij} - a_{ij}| < \varepsilon$ et du fait que $||B - A|| < \varepsilon$, on peut donc majorer nos termes, ce qui donne :

$$|\lambda - a_{ii}| \leq \sum_{i \neq j}^{n} \varepsilon + \sum_{i \neq j}^{n} (|a_{ij}|) + \varepsilon$$

Enfin par un calcul simple de somme, on obtient :

$$|\lambda - a_{ii}| \leq n * \varepsilon + \sum_{i \neq j}^{n} (|a_{ij}|)$$

Ainsi, on obtient une condition sur les pseudo valeurs propres de la matrice A qui va nous permettre d'implémenter notre premier algorithme.

3.2 Pseudo-code

Notre pseudo code se présente sous la forme suivante. Dans la première partie, nous cherchons à créer les disques de Gerschgorin. Puis nous appliquons la condition trouvée précédemment sur tous les points d'une grille afin de trouver ceux qui appartiennent au pseudo spectre. Nous allons expliciter l'algorithme plus en détails dans la partie suivante.

Algorithm 1 Gerschgorin disks's algorithm

procedure PSEUDO-SPECTRUM ▷ creation of Gerschgorin disks
  $[n, m] \leftarrow size(A)$
for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
      S$\leftarrow S + |A(i, j)|$
      R[i]$\leftarrow S$
      if $S > I$ **then**
        $I \leftarrow S$
      end
    end
  end

for $i \leftarrow 1$ **to** n **do**
    $x[i] \leftarrow A[i, i]$
    for $\theta \leftarrow 0$ **to** 2π **do**
      $x \leftarrow R[i] \times cos(\theta) \times x[i]$ ▷ Put the grid
      $y \leftarrow R[i] \times sin(\theta) \times x[i]$
      $J1 \leftarrow min(J1, -R[i] + X[i])$
      $J2 \leftarrow min(J1, R[i] + X[i])$
      $J3 \leftarrow min(J1, -R[i] + X[i])$
      $J4 \leftarrow min(J1, R[i] + X[i])$
      $xh \in]J1; J2[$
      $yv \in]J3; J4[$
      $[X, Y] \leftarrow meshgrid(xh, yv)$
    end
  end
  ▷ seek for points belonging to the pseudo-spectrum
  for $j \leftarrow 1$ **to** $length(xh)$ **do**
    for $i \leftarrow 1$ **to** $length(yv)$ **do**
      $\sigma min[j, k] \leftarrow min(svd((X(j, k) + Y(j, k) \times 1i) \times Id - A)$
    end
  end
  $f \leftarrow [Epsi, Epsi]$ ▷ plot the pseudo-spectrum
end procedure

3.3 Présentation de l'algorithme

Maintenant que nous avons vu la partie théorique du premier algorithme, nous allons pouvoir passer à l'explication du code MATLAB que nous avons réalisé afin de trouver le pseudo-spectre d'une matrice.
Nous avons donc dans un premier temps tracé les disques de Gerschgorin.

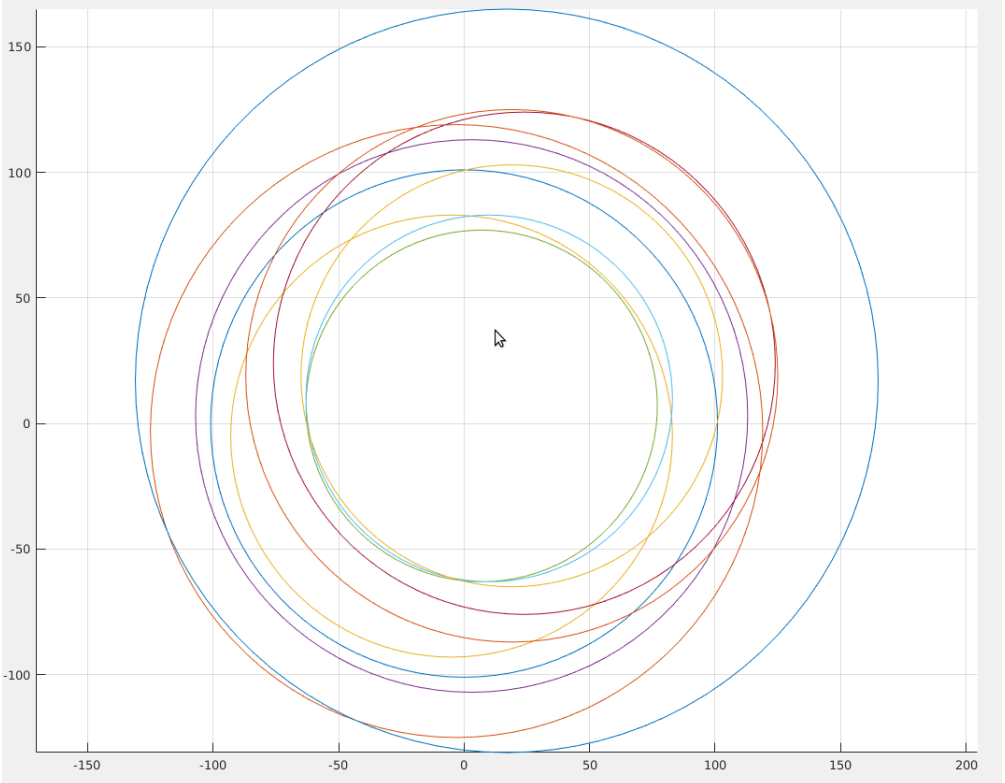


FIGURE 2 – Disques de Gerschgorin à partir d'une matrice aléatoire

Une fois les disques tracés, il a fallu tracer un rectangle qui incluait l'ensemble des disques. Ce dernier permettait de réduire la zone où se trouvait le pseudo spectre.

Puis on supprime les cercles et on trace une grille à l'aide de la commande *meshgrid* (sur MATLAB) dans la zone sélectionnée. Pour cela on a utilisé un pas, que l'utilisateur peut faire varier avec l'utilisation de l'interface graphique. Le pas permet donc de modifier la taille de la grille. Une fois la grille tracée, on étudie point par point pour savoir si ce point correspond au pseudo-spectre.

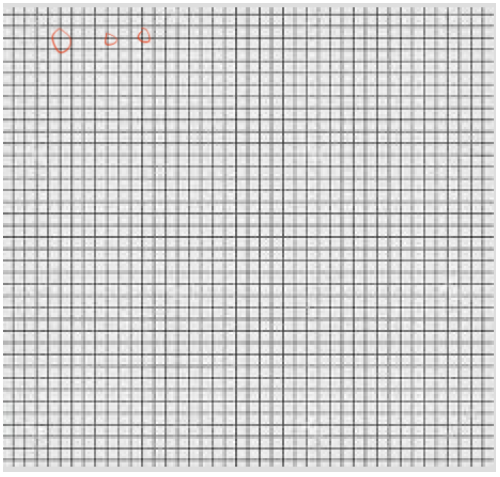


FIGURE 3 – Exemple de la grille obtenue

On vérifiait pour chaque point de la grille la formule suivante : $\sigma_{min}(z * I - A) \leq \varepsilon$ avec z le point de la grille, I la matrice identité, A la matrice choisie au départ, ε fixé par l'utilisateur (qui correspond à la précision souhaitée) et σ_{min} la plus petite valeur singulière.

Si l'inégalité est vérifiée, on conserve le point. Puis on relie tous les points conservés à l'aide la fonction contour sur MATLAB, pour obtenir le pseudo spectre.

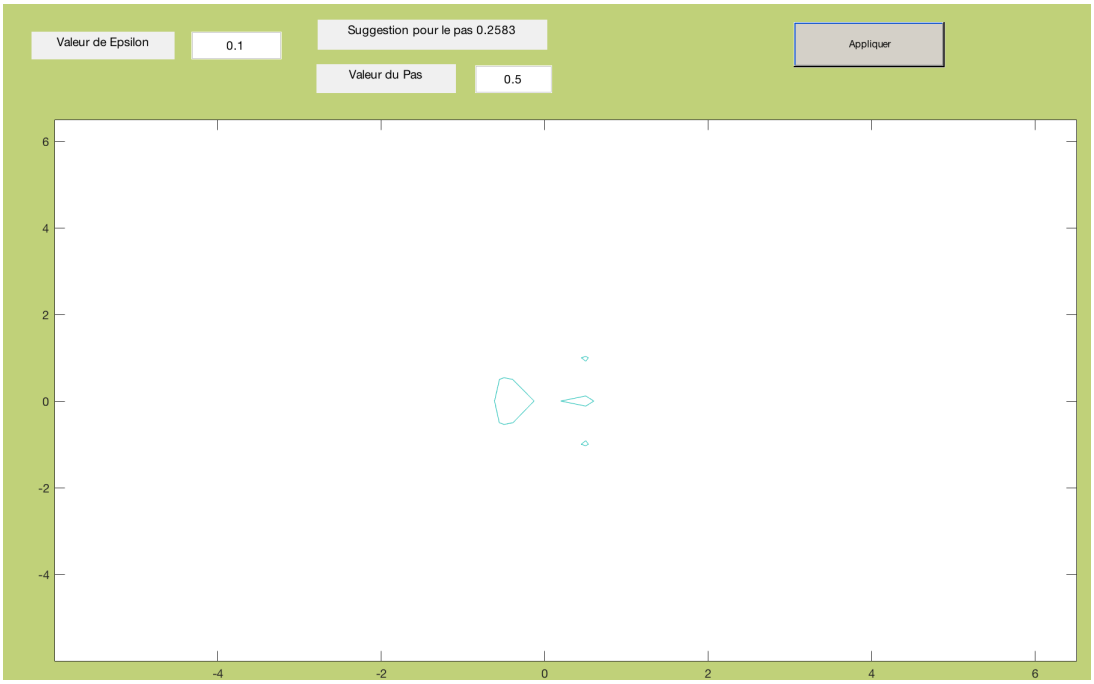


FIGURE 4 – Exemple d'un pseudo spectre obtenu pour une matrice aléatoire A

3.4 Utilisation de l'algorithme

Pour l'algorithme expliqué précédemment nous avons pu faire une interface graphique.

Pour commencer il suffit de rentrer la matrice à étudier sur le terminal de MATLAB, prenons par exemple la matrice A. Ensuite, il ne reste qu'à appeler cette matrice avec notre code : valeurPropre(A).

```
>> A= rand(10)

A =

    0.9304    0.2535    0.5533    0.4834    0.7428    0.0085    0.2108    0.6466    0.7272    0.5718
    0.4470    0.6782    0.2995    0.6848    0.1045    0.4160    0.2654    0.0621    0.0001    0.5929
    0.8339    0.9145    0.8970    0.4800    0.5209    0.4391    0.9932    0.9382    0.1698    0.6986
    0.9878    0.6171    0.7943    0.7465    0.8466    0.7844    0.4808    0.8987    0.4633    0.3058
    0.3696    0.3225    0.7758    0.2114    0.8436    0.8300    0.8278    0.6949    0.3619    0.3939
    0.1708    0.4905    0.5990    0.2485    0.3777    0.5896    0.6032    0.3104    0.9521    0.3369
    0.8232    0.4075    0.5572    0.0978    0.2721    0.1421    0.8178    0.3435    0.9322    0.1290
    0.5871    0.0839    0.2997    0.7087    0.1253    0.5933    0.9555    0.0762    0.9581    0.0869
    0.9616    0.5920    0.5477    0.8557    0.6914    0.7262    0.6504    0.5198    0.2065    0.5489
    0.4891    0.8790    0.9918    0.7890    0.5551    0.4284    0.6276    0.6088    0.1597    0.3177

>> valeurPropre(A)
SetGlobalStep 1
```

FIGURE 5 – Exemple d'application pour une matrice A

Un interface s'affiche alors devant vous. Au tout début, des courbes apparaissent, pour lancer l'algorithme il faut choisir une valeur de epsilon et un pas puis appuyer sur appliquer. Cela vous donnera donc un aperçu du pseudo spectre de votre matrice.

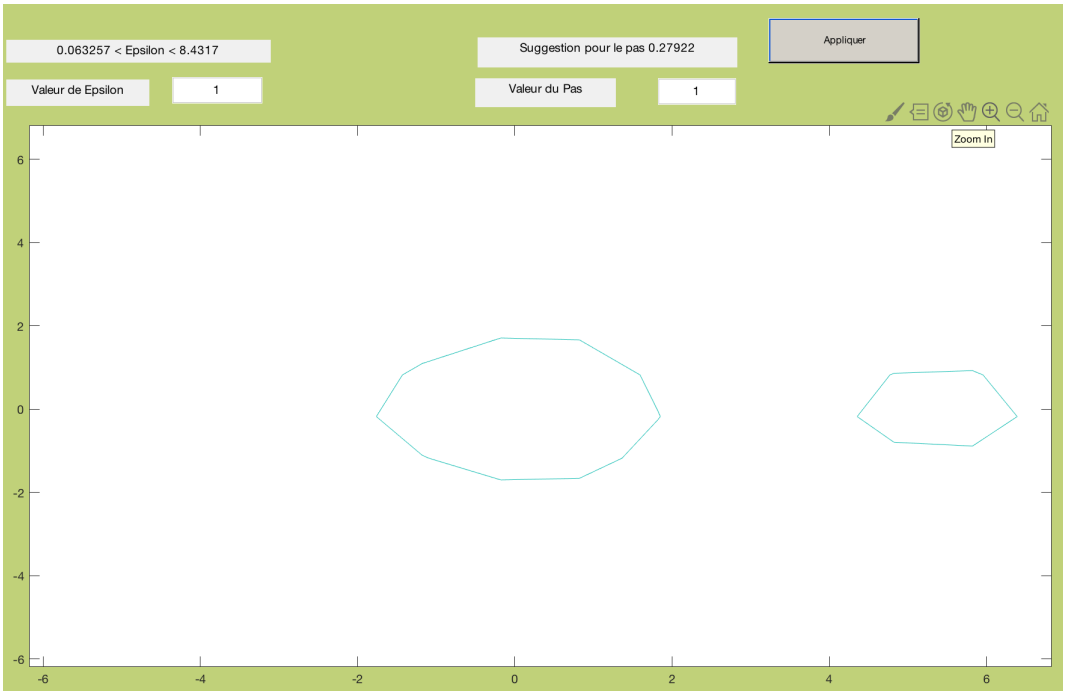


FIGURE 6 – Exemple d'un pseudo-spectre obtenu pour une matrice aléatoire A

Vous pouvez également zoomer pour mieux voir grâce à la petite loupe plus en haut à droite et cela vous permettra d'obtenir un pseudo spectre plus clair.

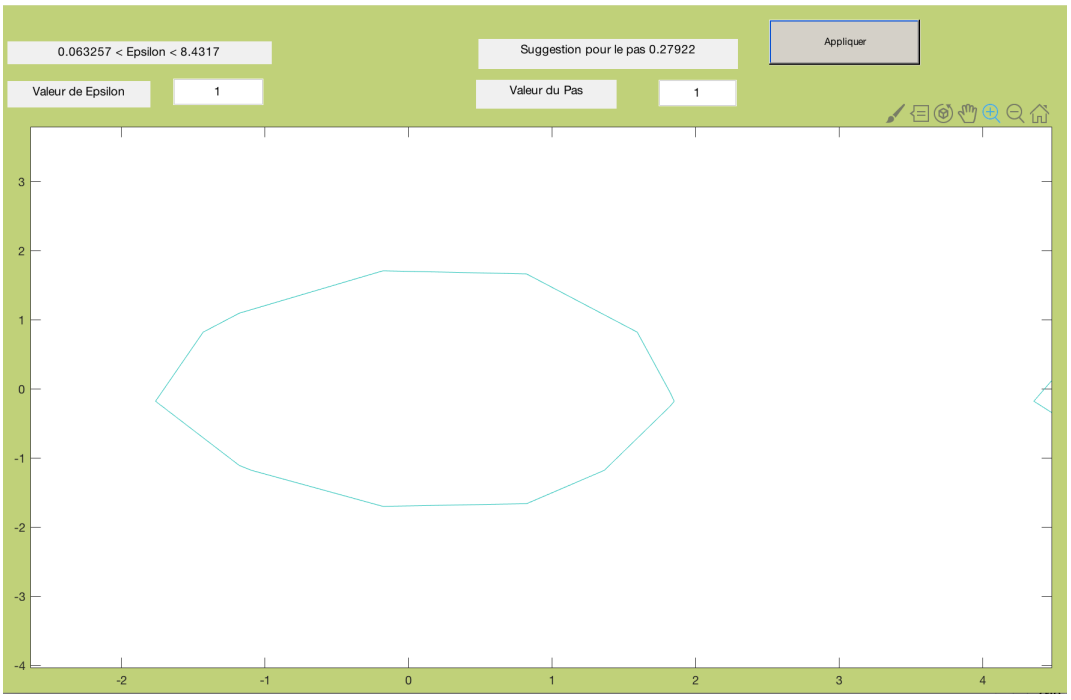


FIGURE 7 – Exemple d'un pseudo-spectre zoom obtenu pour une matrice aléatoire A

Concernant l'interface vous avez du remarquer qu'il y a une suggestion pour les valeurs d'epsilon et du pas.

Avec une valeur d'Epsilon prise au hasard, l'utilisateur peut avoir un écran blanc, laissant à penser que notre code ne fonctionne pas. En fait, la fonction *contour* dessine des points ayant la même valeur. Si aucun point n'a cette valeur, l'écran reste blanc. En ayant le minimum et le maximum de *sigmin*, l'opérateur peut choisir des valeurs utiles.

La valeur du pas détermine la taille de la grille de calcul et donc de la précision mais également du temps de calcul. Le pas suggéré est donc la longueur du rectangle encadrant les cercles de Gerschgorin, divisé par 50 et donne donc une grille de 2500 points. Libre à l'opérateur de diminuer le pas et de suivre le temps de calcul.

Nous avons également rajouter une barre d'avancement pour signaler à l'utilisateur où en est sa demande et le temps qu'il va devoir attendre. Pour cela, on a calculé le nombre de points à traiter en fonction de la grille. Puis on signale à l'utilisateur dès qu'un point est traité. Cela peut prendre plus ou moins de temps en fonction de la taille de la matrice et de la taille de la grille. Il est important de noter que ce genre

d'utilisations sur matlab augmente le temps de calcul.

Une fois cet algorithme fini, nous avons décidé de travailler sur un second algorithme et de comparer les deux.

4 Deuxième algorithme

4.1 Présentation de l'algorithme

L'algorithme utilisé à partir du théorème de Gershgorin est très coûteux car il oblige à tester tous les points du quadrillage pour trouver le pseudo spectre. C'est pourquoi nous avons élaboré un second algorithme basé sur la méthode de Newton.
A partir des valeurs propres de la matrice, on trouve une première valeur du pseudo spectre avec laquelle on applique la méthode de Newton afin de trouver les autres pseudo valeurs propres. Cette méthode consiste à calculer le gradient en un point, puis à trouver le vecteur orthogonal à ce gradient, et enfin à projeter afin trouver la valeur suivante du pseudo spectre.
Nous avons pour cette partie utilisé l'article Martin Brühl : A curve tracing algorithm for computing the pseudospectrum (cité dans la partie source à la fin du rapport).

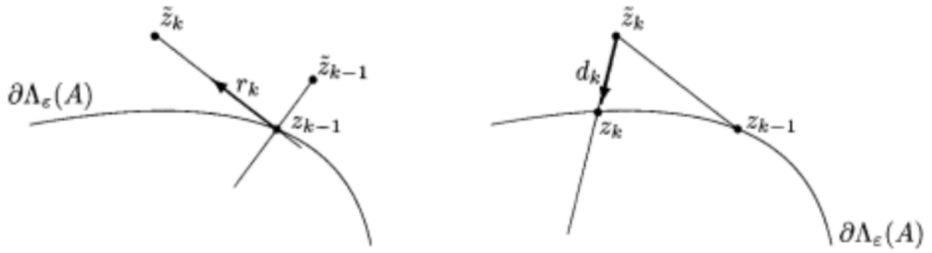


FIGURE 8 – Schéma pour expliquer la méthode de Newton

4.1.1 Initialisation

On sait qu'au moins une valeur propre appartient au pseudo spectre. On trace une droite passant par cette valeur propre d'équation :

$$\lambda_0 + d\theta_0 \tag{1}$$

avec λ_0 la valeur propre pour laquelle on applique l'algorithme et $|d| = 1$ l'orientation de la droite. Cette droite recoupe le bord du pseudo spectre.
Puis on réalise une boucle pour déterminer ce point d'intersection. Tant que ($|\frac{svd(z_1 * I - A - \varepsilon)}{\varepsilon}| > tolérance$) on sait que z_1 n'appartient pas au contour du pseudo spectre et on applique la méthode de Newton à partir de l'équation :

$$z_1 = \lambda_0 + \theta_1 * d$$

avec

$$\theta_1 = -\frac{\sigma_{min} - \varepsilon}{\Re(\bar{d}u^*_{min}v_{min})} \tag{2}$$

et on a que u^*_{min} est la transconjuguée de u_{min}.

Après avoir trouvé la première valeur propre, on exécute les étapes de prédiction et de correction, toujours à partir de la méthode de Newton. Ces deux étapes consistent à tracer le gradient du point trouvé, puis projeter ce gradient afin de retomber sur une valeur du pseudo-spectre qui sera notre point suivant.

4.1.2 Prédiction

Nous calculons le gradient avec la formule suivante :

$$gradient = \frac{iv^*_{min}u_{min}}{|v^*_{min}u_{min}|}. \tag{3}$$

et on a que v^*_{min} est la transconjuguée de v_{min}.
De plus pour v_{min} et u_{min}, on les calcule grâce aux SVD du premier point trouvé.
Ensuite, on fixe le pas τ_k à 0.1. Puis on calcule un $\widetilde{z_k}$ à l'aide de la formule suivante :

$$\widetilde{z_k} = z_k - 1 + \tau_k \times gradient \tag{4}$$

4.1.3 Correction

On recalcule les trois matrices (u_{min}, v_{min},σ_{min}) obtenues grâce à la SVD de $\widetilde{z_k} \times I - A$.
Avec I la matrice identité, A la matrice étudiée et $\widetilde{z_k}$ le point calculé précédemment.
Pour finir on trouve le point suivant grâce à l'équation ci-dessous :

$$z_k = \widetilde{z_k} - \frac{\sigma_{min} - \varepsilon}{u^*_{min}v_{min}} \tag{5}$$

On applique l'ensemble pour toutes les valeurs propres de la matrice et on affiche les points trouvés.

4.2 Pseudo-code

Nous nous basons sur les explications précédentes pour définir un algorithme qui implémente la méthode de Newton. La première boucle permet de trouver la première valeurs propres. Nous avons ensuite une boucle while qui permet d'appliquer la partie prédiction de la méthode de Newton, puis une boucle qui permet d'appliquer la méthode de correction. Afin de réaliser cet algorithme pour l'ensemble des valeurs propres, on inclut l'ensemble de ces boucles dans une boucle for qui parcourt les valeurs propres de la matrice.

Algorithm 2 Newton's method

procedure PREDICTION-CORRECTION(A)
$tol = 0.1$
$N = 1000$
$\varepsilon = 0.1$
$I = I_n$
for $\lambda \leftarrow 1$ **to** $length(A)$ **do**
$z_0 \leftarrow \lambda + \varepsilon$
while $\frac{|\sigma_{min}(z_0 * I - A) - \varepsilon|}{\varepsilon}$ **do**
$z_0 = z_0 - \frac{\sigma_{min} - \varepsilon}{Re(v_{min} * u_{min})}$
end
$z_j = z_0$
while $j < N$ **do**
▷ determine the direction r and length of the step τ
$gradient = \frac{i * v_{min} * u_{min}}{|v_{min} * u_{min}|}$
$\tau_k = 0.1$
$z_{j_t} = z_j + \tau_k * gradient$
while $|min(svd(z_{j_t} * I - A))| - \varepsilon > tol * \varepsilon$ **do**
$z_{j_t} = z_{j_t} - \frac{\sigma_{min} - \varepsilon}{u_{min} * v_{min}}$
end
$z_j = z_{j_t}$ ▷ display point
$j = j + 1$
end
end

end procedure

4.3 Utilisation de l'algorithme

L'utilisation de l'algorithme est très simple, il suffit de rentrer une matrice aléatoire ou non dans le terminal de MATLAB comme ceci par exemple :

```
> B=rand(10)

 =

   0.8147    0.1576    0.6557    0.7060    0.4387    0.2760    0.7513    0.8407    0.3517    0.0759
   0.9058    0.9706    0.0357    0.0318    0.3816    0.6797    0.2551    0.2543    0.8308    0.0540
   0.1270    0.9572    0.8491    0.2769    0.7655    0.6551    0.5060    0.8143    0.5853    0.5308
   0.9134    0.4854    0.9340    0.0462    0.7952    0.1626    0.6991    0.2435    0.5497    0.7792
   0.6324    0.8003    0.6787    0.0971    0.1869    0.1190    0.8909    0.9293    0.9172    0.9340
   0.0975    0.1419    0.7577    0.8235    0.4898    0.4984    0.9593    0.3500    0.2858    0.1299
   0.2785    0.4218    0.7431    0.6948    0.4456    0.9597    0.5472    0.1966    0.7572    0.5688
   0.5469    0.9157    0.3922    0.3171    0.6463    0.3404    0.1386    0.2511    0.7537    0.4694
   0.9575    0.7922    0.6555    0.9502    0.7094    0.5853    0.1493    0.6160    0.3804    0.0119
   0.9649    0.9595    0.1712    0.0344    0.7547    0.2238    0.2575    0.4733    0.5678    0.3371

> mac10(B)
```

FIGURE 9 – Exemple d'utilisation du terminal

Puis on on lance le programme et obtient une figure. Pour la matrice donnée dans l'exemple on obtient :

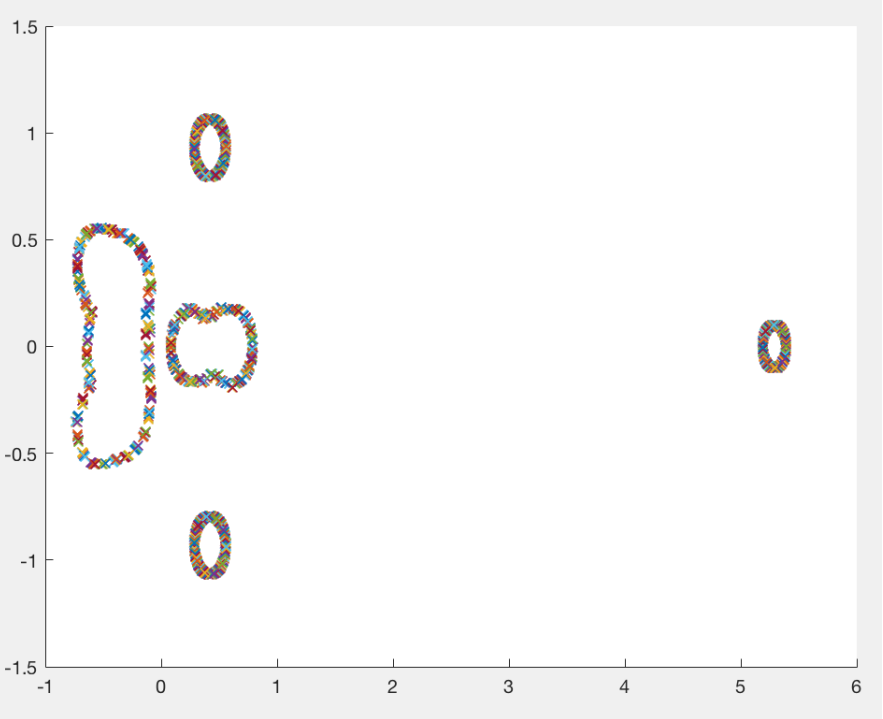


FIGURE 10 – Exemple d'un pseudo spectre obtenu pour une matrice aléatoire

5 Comparaison des deux algorithmes

Le but de ce projet était donc de réaliser deux algorithmes pour obtenir des pseudo-spectres, de comprendre la notion de pseudo-spectre et de découvrir son utilité.
Comme expliqué précédemment l'algorithme de Gershgorin est très coûteux par rapport à celui basé sur la méthode de Newton.
Toute fois, nous obtenons des représentation graphique très similaires, à partir de la même matrice, comme vous pouvez le voir sur les figures ci-dessous.

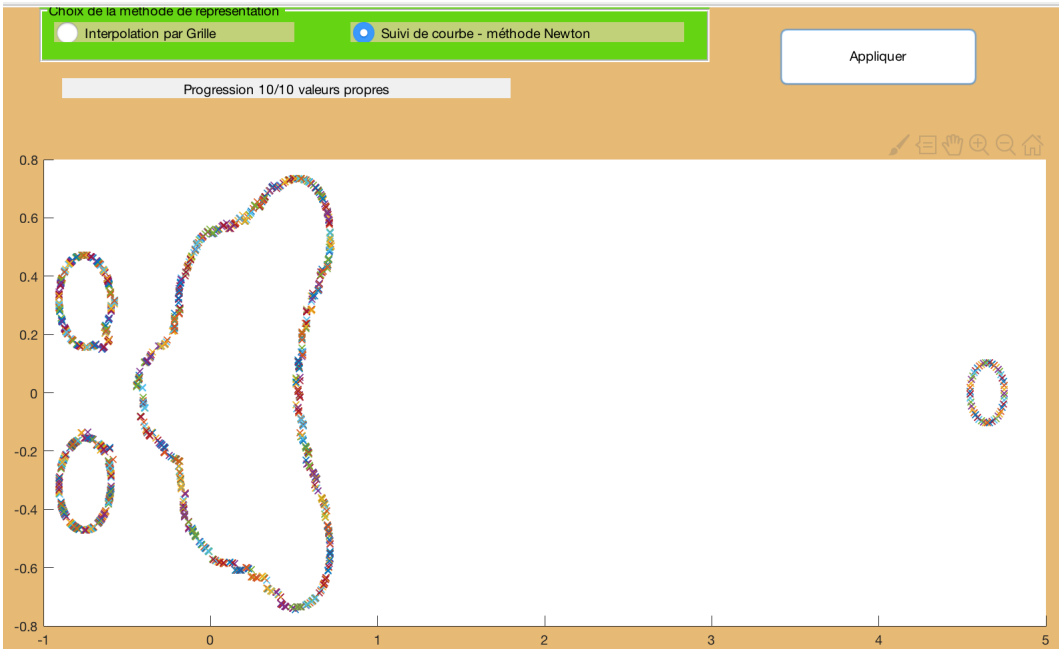


FIGURE 11 – Exemple d'un pseudo-spectre obtenu pour une matrice à partir de la méthode de Newton

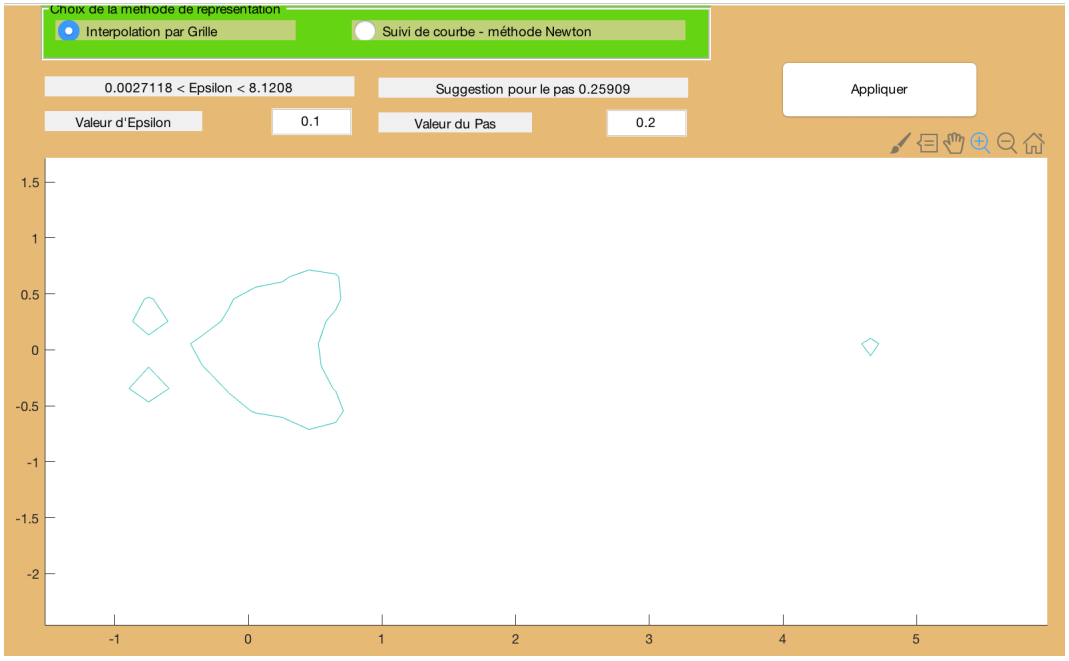


FIGURE 12 – Exemple d'un pseudo-spectre obtenu pour une matrice à partir de la méthode de Gershgorin

Une fois, nos deux algorithmes finis, nous avons modifié l'interface graphique pour que l'utilisateur puisse avoir le choix entre les deux programmes.
Il est important de sélectionner en premier Newton pour débloquer l'interface et après libre à l'utilisateur de commencer par celui qu'il préfère ou de faire les deux.

FIGURE 13 – Exemple de l'interface graphique pour la méthode de Newton

FIGURE 14 – Exemple de l'interface graphique pour la méthode de Gershgorin

L'utilisation est encore une fois très simple. Il suffit de rentrer la matrice que l'on veut étudier dans le terminal de MATLAB (par exemple A), puis d'appeler le programme PseudoSpectre(A). Comme sur le schéma ci-dessus, une interface graphique s'ouvre devant vous. Il vous faut donc choisir la méthode de votre choix et appuyer sur appliquer.
Concernant la méthode de Newton vous pouvez choisir l'epsilon qui vous convient et pour la méthode de Gershgorin vous pouvez encore choisir le pas et l'epsilon. N'hésitez pas à utiliser la loupe pour zoomer ou dé-zommer.

6 Stabilité et utilité des pseudo-spectres

L'étude du pseudo spectre a débuté avec le constat que la connaissance seule du spectre d'une matrice fait apparaître des différences entre la théorie et la pratique, et ne permet donc pas d'avoir toutes les informations contenues dans le spectre.
Afin d'étudier le pseudo-spectre d'une matrice, il faut réaliser des perturbations sur cette matrice. Un point appartient au pseudo-spectre d'une matrice si et seulement si ce point appartient au spectre d'une perturbation $A + E$ de cette matrice avec $||E|| < \varepsilon$. Les différentes lignes de niveau du pseudo spectre correspondent aux différentes perturbations en fonction de la valeur de ε. Nous nous sommes limitées à l'étude d'une ligne de niveau pour une seule valeur de ε.
L'ensemble des valeurs propres se trouve à l'intérieur des lignes de niveau du pseudo spectre. L'étude des pseudo-spectres est très importante et est beaucoup utilisée par les physiciens. On la retrouve notamment en mécanique des fluides où la stabilité des systèmes étudiés est déterminée par la position du pseudo spectre par rapport à l'axe des abscisses. Mais seulement quelques modèles unidimensionnels, comme par exemple l'oscillateur harmonique, sont bien compris. Certains chercheurs ont également établi que le fait qu'une matrice soit normale, sous de petites perturbations, assure la stabilité de l'ensemble de ses valeurs propres. Il reste encore beaucoup à apprendre sur le comportement du spectre en pratique, et nul doute que de futurs problèmes sur le sujet se poseront à nous dans différents domaines.

7 Conclusion

Pour conclure, ce projet nous a permis de comprendre la notion de pseudo-spectre et de pseudo valeur propre. Nous avons pu également mettre en pratique certaines notions apprises durant notre année scolaire comme l'analyse matricielle. Nous nous sommes rendus compte que l'étude des pseudo-spectre est un outil majeur pour la communauté scientifique et nous sommes très heureuses d'avoir eu un aperçu et d'avoir pu étudier certains algorithmes permettant de le déterminer. De plus, nous avons pu travailler sur MATLAB, ce qui nous a donné l'opportunité de mieux maîtriser ce logiciel. Nous avons également consolider nos compétences à travailler en groupe et dans un délais donné.
Nous n'avons pas pu directement étudier la stabilité d'un pseudo-spectre car cela est plutôt fait par les physiciens. Notre but durant ce projet était donc de fournir un outils mathématiques pour certains spécialistes qui pourront alors l'interpréter.

8 Sources

Nous avons fait des recherches concernant les notions théoriques en nous basant sur certains cours ou des livres.
Vous trouverez les références ci-dessous :

- **Analyse Spectrale pour les Systèmes Dynamiques Classiques et Quantiques :** `http://perso.utinam.cnrs.fr/~viennot/doc/coursNLCC.pdf`
- **Spectres et Pseudo-spectres :** `https://www.math.univ-toulouse.fr/~jroyer/TD/2013-14-Projet-L3/Projet-Blohorn-Boquen-Dubois.pdf`
- **Thèse : Etude du pseudo-spectre d'operateurs non auto-adjoints :** `https://tel.archives-ouvertes.fr/tel-00109895/document`
- **Martin Brühl, A curve tracing algorithm for computing the pseudospectrum, BIT,36(3) :441–454, 1996**
- A. N. Malyshev and M. Sadkane, **Componentwise pseudospectrum of a matrix.LinearAlgebra Appl, 378 :283–288, 2004**
- Lloyd N. Trefethen and Mark Embree, **Spectra and pseudospectra. Princeton Univer-sity Press, Princeton, NJ, 2005**
- **Cours d'analyse matricielle :** `https://moodle-sciences.upmc.fr/moodle-2018/pluginfile.php/450448/mod_resource/content/1/main-MAM_1.pdf`

9 Annexes

9.1 Déroulement des séances

Pour cette étude nous sommes une équipe de trois étudiantes, encadrées par un tuteur : Mr Stef Grallat. Nous disposons d'une séance hebdomadaire le mercredi matin qui nous permettait de travailler en groupe, avec notre tuteur et d'avancer ensemble.

Vendredi 22/02/19 :	Présentations des différents projets et choix des groupes.
Mercredi 27/02/19 :	Prise de contact avec l'encadrant : explication du projet et des objectifs.
Mercredi 06/03/19 :	Début des recherches, prise en main de MATLAB et début de la démonstration (de Gershgorin).
Mercredi 13/03/19 :	Codage du premier algorithme concernant les disques de Gershgorin et recherches théoriques et démonstration.
Mardi 20/03/19 :	Codage premier algorithme.
Mercredi 27/03/19 :	Validation du premier algorithme et création de l'interface graphique.
Mercredi 03/04/19 :	Finition de l'interface graphique.
Mercredi 10/04/19 :	Explication du deuxième algorithme basé sur la méthode de Newton.
Mercredi 17/04/19 :	Recherche et travail de lecture sur le deuxième algorithme.
Semaines de vacances :	Codage du second algorithme.
Jeudi 09/05/19 :	Premier rapport et continuation du second algorithme.
Mercredi 15/05/19 :	Fin du deuxième algorithme.
Mercredi 22/05/19 :	Fin du rapport et création de la nouvelle interface regroupant les deux algorithmes.
Mercredi 29/05/19 :	Répétition oral et révision des transparents.
Mercredi 05/06/19 :	Soutenance.

Nous avons rajouté des séances au cours de l'année lorsque nos emplois du temps nous le permettaient. Nous travaillions également de manière individuelle chaque semaine pour mettre en commun nos avancées le mercredi. Chaque semaine nous avions un rendez vous avec notre tuteur et nous faisions le point de nos avancées, de nos objectifs et des erreurs.

9.2 Codes

De manière à rendre nos codes plus lisibles et plus faciles d'accès nous avons créé un github : https ://github.com/GMorand/Projet-initiation.git Toutefois, les interfaces

graphiques peuvent parfois ne pas fonctionner sous certaines versions de Matlab. Une démonstration sera faite durant la soutenance.