

# Calcul de pseudospectres

Ahmed Akram Bouaziz, Ron Sharony  
Sous la direction de Stef Graillat

2024

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Définitions</b>	<b>2</b>
2.1	Définitions du pseudospectre . . . . .	2
<b>3</b>	<b>GRID</b>	<b>2</b>
3.1	Description générale . . . . .	2
3.2	Théorème de Gershgorin . . . . .	2
3.3	Disques de Gershgorin généralisés pour le pseudospectre . . . . .	3
3.4	Pseudo-code de GRID . . . . .	3
3.5	Exemple . . . . .	4
3.6	Analyse de l'algorithme . . . . .	4
<b>4</b>	<b>Prédiction-correction</b>	<b>4</b>
4.1	Préliminaire : la méthode de Newton . . . . .	4
4.2	Description générale . . . . .	4
4.3	Pseudo-code de prédiction-correction . . . . .	5
4.4	Exemple . . . . .	6
4.5	Analyse de l'algorithme . . . . .	6
<b>5</b>	<b>Comparaison de performances</b>	<b>6</b>
<b>6</b>	<b>Abscisse et rayon du pseudospectre</b>	<b>7</b>
6.1	Motivation . . . . .	7
6.2	Abscisse du pseudospectre : algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$ . . . . .	7
6.3	Illustration de l'algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$ . . . . .	8
6.4	Pseudo-code de l'algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$ . . . . .	8
<b>7</b>	<b>Pseudospectre par composantes</b>	<b>9</b>
7.1	Introduction . . . . .	9
7.2	Pseudo-code du calcul du pseudospectre par composantes . . . . .	9
7.3	Exemple . . . . .	10
<b>8</b>	<b>Conclusion</b>	<b>10</b>

## 1 Introduction

Le  $\varepsilon$ -pseudospectre d'une matrice  $A$  est défini comme le sous-ensemble du plan complexe consistant en toutes les valeurs propres de toutes les matrices situées à une distance  $\varepsilon$  de  $A$ . C'est un outil très utilisé en théorie du contrôle et en automatique pour tester la robustesse de la stabilité d'un système. Pour l'étudier, nous verrons dans un premier temps une méthode, connue sous le nom de GRID, qui par une caractérisation du pseudospectre comme ensemble de valeurs singulières permet de facilement le calculer. En revanche, cet algorithme nécessite beaucoup de calculs de valeurs singulières, notamment en de nombreux points qui ne font pas partie du pseudospectre. C'est pourquoi nous utilisons ensuite un algorithme de prédiction-correction pour pallier ce problème, se basant en grande partie sur la méthode de Newton : méthode efficace pour trouver une approximation d'un zéro d'une fonction.

## 2 Définitions

### 2.1 Définitions du pseudospectre

Il existe plusieurs définitions du  $\varepsilon$ -pseudospectre d'une matrice  $A$  noté  $\sigma_\varepsilon(A)$  qui sont à l'origine d'algorithmes permettant de le calculer que nous allons énoncer. D'abord les définitions que nous utiliserons,

**Définition 1** ([TE05]).

$$\sigma_\varepsilon(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\| > \varepsilon^{-1}\}$$

ou bien si  $\|\cdot\| = \|\cdot\|_2$ ,

**Définition 2.**

$$\sigma_\varepsilon(A) = \{z \in \mathbb{C} : s_{\min}(zI - A) < \varepsilon\}$$

où  $s_{\min}(\cdot)$  est la plus petite valeur singulière (les valeurs singulières étant toutes les racines carrées d'une valeur propre de  $A^*A$ ).

Mais d'autres définitions équivalentes, comme celle ci-dessous, sont à l'origine d'autres algorithmes de calcul de pseudospectres que nous ne traiterons pas ici.

**Définition 3.**

$$\sigma_\varepsilon(A) = \bigcup_{\|\mathbf{E}\| < \varepsilon} \sigma(A + \mathbf{E}),$$

où  $\sigma(\cdot)$  est le spectre et  $E$  est une matrice quelconque de perturbation telle que  $\|\mathbf{E}\| < \varepsilon$ .

## 3 GRID

### 3.1 Description générale

En utilisant la deuxième définition du pseudospectre, on peut utiliser un algorithme naturel pour tracer le pseudospectre. Il suffit de calculer les valeurs de  $s_{\min}(zI - A)$  pour tout  $z$  dans une grille souhaitée et tracer le contour. Pour obtenir la grille, il nous faut un encadrement du pseudospectre. Pour cela, on généralise les disques de Gershgorin.

### 3.2 Théorème de Gershgorin

Soit  $A \in M_n(\mathbb{C})$ .

**Définition 4.** Le centre  $C_i$  et le rayon  $R_i$  du disque de Gershgorin associé à la  $i$ -ème ligne de la matrice  $A$  sont définis par :

$$C_i = a_{ii} \quad \text{et} \quad R_i = \sum_{j \neq i} |a_{ij}|, \quad 1 \leq i \leq n.$$

On définit le  $i$ -ème disque de Gershgorin  $D_i$  par

$$D_i = \left\{ z \in \mathbb{C} \mid |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|, 1 \leq i \leq n \right\}.$$

**Théorème 3.1.** Toute valeur propre de  $A$  appartient au moins à l'un des disques de Gershgorin.

*Démonstration.* Soient  $\lambda$  une valeur propre de  $A$  et  $X = (x_1, \dots, x_n)$  un vecteur propre associé. Pour tout  $i$  compris entre 1 et  $n$ , on a  $(AX)_i = (\lambda X)_i$ .

D'où  $(\lambda - a_{ii})x_i = \sum_{j \neq i} a_{ij}x_j$ .

Choisissons un indice  $i_{\max}$  pour lequel le module de  $x_{i_{\max}}$  est maximal. Puisque  $X$  est un vecteur propre,  $|x_{i_{\max}}|$  est non nul et par inégalité triangulaire on a donc

$$|a_{ii} - \lambda| = \left| \sum_{j \neq i} a_{ij} \frac{x_j}{x_i} \right| \leq \sum_{j \neq i} \left| a_{ij} \frac{x_j}{x_i} \right| \leq \sum_{j \neq i} |a_{ij}| = R_i.$$

□

### 3.3 Disques de Gershgorin généralisés pour le pseudospectre

Dans cette partie on utilise la troisième définition du  $\sigma_\varepsilon$  pseudospectre.

**Théorème 3.2.** *Avec les notations précédentes, on a :*

$$\sigma_\varepsilon \subset \bigcup_{i=1}^n D(C_i, R_i + \sqrt{n}\varepsilon).$$

*Démonstration.* Soit  $E \in M_n(\mathbb{C})$  une matrice de perturbation telle que  $\|E\|_2 < \varepsilon$ .

Soient  $A \in M_n(\mathbb{C})$  et  $C_i, R_i, i \in \{1, \dots, n\}$  les centres et les rayons des disques de Gershgorin de  $A$ .

Alors pour la matrice  $M = A + E$ , on a  $\|M - A\|_\infty = \|E\|_\infty$ .

L'inégalité sur les normes induites  $\|\cdot\|_\infty$  et  $\|\cdot\|_2$  sur  $M_n(\mathbb{C})$  :  $\|A\|_\infty \leq \sqrt{n}\|A\|_2$  découle de l'équivalence des normes et de l'inégalité sur les normes correspondantes sur  $\mathbb{C}^n$ .

Soient  $R'_i, C'_i$  les rayons et centres des disques de Gershgorin de  $M$ , on pose  $S := \sum_{j=1, j \neq i}^n |(m - a)_{ij}|$ .

On a  $S + |(m - a)_{ii}| = \|M - A\|_\infty = \|E\|_\infty \leq \sqrt{n}\varepsilon$ .

De plus, on a  $R'_i - R_i \leq S$  et  $|C'_i - C_i| = |(m - a)_{ii}| = \|M - A\|_\infty - S \leq \sqrt{n}\varepsilon - S \leq \sqrt{n}\varepsilon - S$ .

Pour  $z \in \sigma(A + E)$ , on a  $|z - C'_i| \leq R'_i \leq R_i + S$ .

Donc  $|z - C_i| \leq |z - C'_i| + |C'_i - C_i| \leq R_i + S + \sqrt{n}\varepsilon - S = R_i + \sqrt{n}\varepsilon$  par l'inégalité triangulaire.

Donc on a  $\sigma(A + E) \subset \bigcup_{i=1}^n D(C_i, R_i + \sqrt{n}\varepsilon)$  pour tout  $E \in M_n(\mathbb{C})$  telle que  $\|E\|_2 < \varepsilon$ .

D'où, enfin,  $\sigma_\varepsilon \subset \bigcup_{i=1}^n D(C_i, R_i + \sqrt{n}\varepsilon)$ . □

Finalement, la grille est uniformément divisée, chaque carré circonscrivant les disques de Gershgorin contenant  $m$  points le long de sa longueur de sa hauteur avec  $m$  donné.

### 3.4 Pseudo-code de GRID

Soit  $A \in M_n(\mathbb{C})$ . On note  $m$  le nombre de points en lesquels on calcule  $s_{\min}$ , points dont les coordonnées  $(x, y)$  sont inscrits dans des tableaux  $X$  et  $Y$ . Ces  $m$  points sont pris aléatoirement dans  $n$  les disques de Gershgorin adaptés. On note  $S_{\min}$  le tableau dans lequel figurent les valeurs singulières minimales calculées. On trace le contour de  $S_{\min}$  (selon la courbe de niveau  $C = \varepsilon$ ) et on obtient un encadrement de  $\sigma_\varepsilon(A)$ .

---

#### Algorithme 1 GRID

---

```

pour  $i = 0$  à  $n - 1$  faire
  pour  $j = 0$  à  $m - 1$  faire
    pour  $k = 0$  à  $m - 1$  faire
       $S_{\min}[j, k] = s_{\min}(A, n, X[j, k], Y[j, k])$ 
    fin pour
  fin pour
  contour( $X, Y, S_{\min}, \varepsilon$ )
fin pour

```

---

### 3.5 Exemple

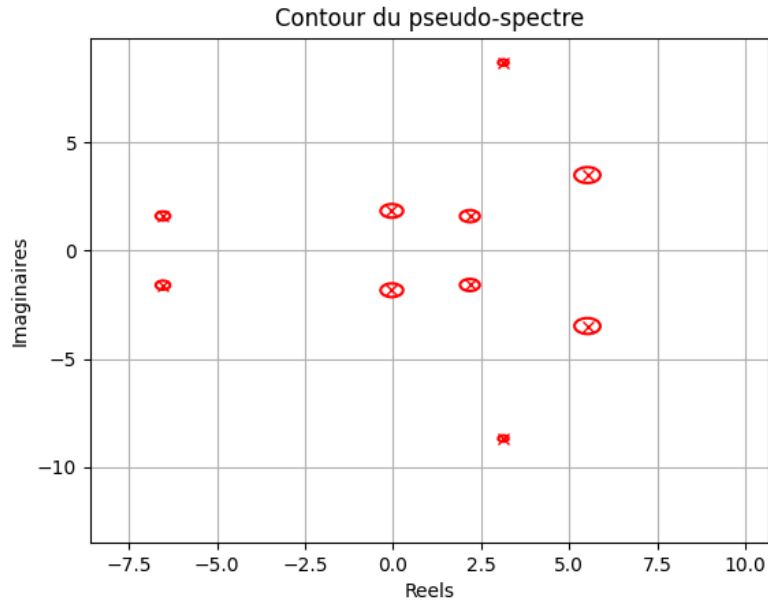


FIGURE 1 – Pseudospectre tracé par GRID pour une matrice de taille 10,  $\varepsilon = 0.1$  et 500 points

### 3.6 Analyse de l'algorithme

Pour  $\varepsilon$  pas trop petit et  $n$  petit, cet algorithme marche bien, mais puisque le coût des calculs des  $s_{\min}$  est assez élevé, et le fait qu'un epsilon plus petit nécessite un grillage plus fin (et donc plus de calculs), le temps d'exécution devient très lent, et on a moins de précisions sur la forme du pseudospectre. Cela est encore pire lorsqu'on augmente  $n$  parce que le nombre de disques de Gershgorin, et on a donc plus de composantes à calculer.

## 4 Prédiction-correction

### 4.1 Préliminaire : la méthode de Newton

La méthode de Newton est un algorithme efficace pour trouver une approximation précise d'un zéro d'une fonction  $f$ . En partant d'un point initial  $x_0$  choisi (idéalement) proche du zéro recherché, on considère  $f$  égale à sa tangente en ce point (DL de premier ordre) :  $f(x) = f(x_0) + f'(x_0)(x - x_0)$  puis pour trouver un zéro de cette approximation on regarde son intersection avec l'axe des abscisses i.e  $0 = f(x_0) + f'(x_0)(x - x_0)$ . On obtient ainsi le point suivant  $x_1$  plus proche du zéro recherché. Itérativement, on a alors  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ . On arrête le procédé itératif selon une certaine tolérance d'erreur, ou bien lorsqu'il n'y a pas convergence (si on n'a pas la dérivabilité de  $f$  en les points considérés ou bien si la tangente à la courbe coupe l'axe des abscisses en un point en dehors du domaine de définition de  $f$ ).

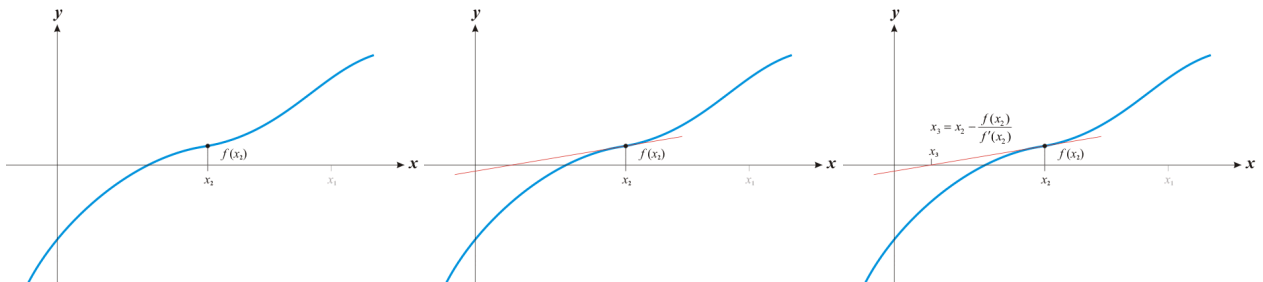


FIGURE 2 – Une itération de la méthode de Newton

### 4.2 Description générale

L'algorithme de GRID étant assez naïf, il est naturel qu'il soit si inefficace, mais il y a plusieurs manières de l'améliorer. Dans cette partie, on s'intéresse à réduire le nombre de points sur lesquels on calcule  $s_{\min}$ .

Pour cela, on utilise l'algorithme de prédiction-correction, qui nous donne une manière de tracer le pseudospectre en faisant des calculs seulement sur des points du contour.

L'idée est de calculer un premier point sur le contour du pseudospectre, de choisir une direction tangente au contour, de se déplacer (c'est-à-dire de prédire le prochain point sur lequel on effectuera le calcul de la plus petite valeur singulière) et puis de projeter (donc corriger, entre autres, le point prédit) sur le contour.

Avant de donner plus de précisions pour chaque étape que l'on vient de citer, il faut garder en tête que l'essentiel de cette approche est le fait que l'on cherche les solutions de  $s_{\min}(zI - A) = \varepsilon$  le long de l'ensemble des  $\lambda + \theta * d$  dans le plan complexe. On pose  $g(z) = s_{\min}(zI - A)$  et d'après [Brü96] on a que  $g$  est dérivable en  $\lambda$  et que  $\nabla g(\lambda) = \mathbf{v}_{\min}^* \mathbf{u}_{\min} \neq 0$  où  $*$  est la transconjuguée et  $\mathbf{u}_{\min}, \mathbf{v}_{\min}$  sont les vecteurs associés à la plus petite valeur singulière. On utilisera la méthode de Newton pour corriger les points prédits.

Précisons rapidement l'étape du calcul du premier point qui utilise la méthode de Newton : le point  $x_0$  choisi est une valeur propre de notre matrice (logiquement proche du zéro recherché). A partir de celui-ci, on utilise la méthode de Newton en suivant une direction (dans notre cas  $d = i$ ) afin d'arriver, (arbitraire, on introduit une tolérance d'erreur) à un point du contour. Pour trouver le premier point, de nombreux calculs de  $s_{\min}$  sont effectués mais à partir de ce point, on aura pour objectif de ne calculer qu'une fois le triplet  $(s_{\min}, \mathbf{u}_{\min}, \mathbf{v}_{\min})$  par point du contour.

On prédit ensuite le prochain point du contour (en suivant une direction indiquée par  $\nabla g$  normalisé et à l'aide d'un pas arbitraire). C'est en ce point que l'on effectue le calcul du triplet  $(s_{\min}, \mathbf{u}_{\min}, \mathbf{v}_{\min})$  puisque corriger le point requiert des informations sur l'imprécision du point prédit. L'étape de correction ne consiste alors qu'en une seule itération de la méthode de Newton, puisque l'on cherche un zéro de la fonction  $h(z_{\text{predict}}) = g(z_{\text{predict}}) - \varepsilon$ . On itère ce processus jusqu'à obtenir une condition d'arrêt (soit en fixant un nombre de points à calculer, soit en calculant des points jusqu'à ce que l'on revienne au premier, selon une tolérance arbitraire). Illustrons le passage d'un point  $z_{k-1}$  du contour au point suivant  $z_k$ .

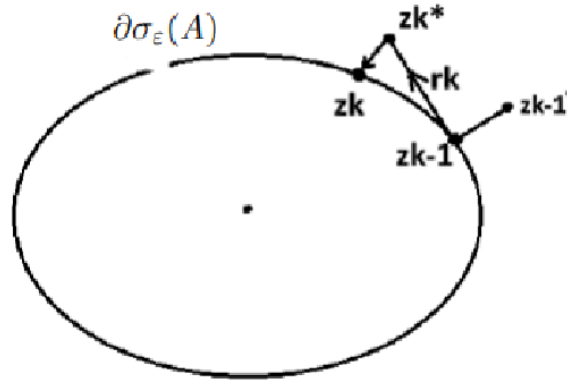


FIGURE 3 –  $k$ -ième itération de l'algorithme.  $z_{k-1}^*$  est le dernier point prédit

A partir de  $z_{k-1}$ , on cherche à trouver un point proche  $z_k$  de  $\partial\sigma_\varepsilon(A)$  (la bordure du pseudospectre). Pour se faire, avec  $g(z) = s_{\min}(zI - A) = s_{\min}((x + iy)I - A)$  on a que

$$\nabla g(x, y) = \begin{pmatrix} \text{Re}(\mathbf{v}^* \mathbf{u}) \\ \text{Im}(\mathbf{v}^* \mathbf{u}) \end{pmatrix},$$

où  $\mathbf{u}$  et  $\mathbf{v}$  sont les vecteurs singuliers correspondant à la valeur singulière minimale de  $((x + iy)I - A)$ . Dans l'optique de suivre la courbe de niveau pour trouver le point le plus proche  $z_k$  de  $\partial\sigma_\varepsilon(A)$ , on cherche orthogonalement à la direction de descente la plus rapide («steepest descent»). Comme celle-ci est  $\mathbf{v}^* \mathbf{u}$ , on suit la direction  $i\mathbf{v}^* \mathbf{u}$ . Après avoir utilisé la prédiction selon cette direction, on corrige par une itération de la méthode de Newton. D'où le pseudo-code qui suit.

### 4.3 Pseudo-code de prédiction-correction

Soient  $A \in M_n(\mathbb{C}), \varepsilon > 0, \lambda$  valeur propre de  $A$ , un pas  $\tau$ , une tolérance  $tol$  d'erreur par rapport au premier point et une tolérance  $tol_{\text{contour}}$  quant à l'arrêt.

```

pour  $i = 0$  à  $n - 1$  faire
  ÉTAPE 0 : calcul du premier point  $z_1 \in \partial\sigma_\varepsilon(A)$ 
   $\theta_0 = \varepsilon, z = \lambda + i * \theta_0$ 
  tant que  $|g(z) - \varepsilon|/\varepsilon > tol$  faire
    mise à jour de  $z$  selon l'équation  $z = z - \frac{s_{\min} - \varepsilon}{\operatorname{Re}(-iv_{\min}^* u_{\min})} i$ 
  fin tant que
   $z_1 = z$ 
  tant que  $|z_k - z_1| > tol_{\text{contour}}$  faire
    ÉTAPE 1 : Prédiction
     $r_k = i \frac{\mathbf{v}^* \mathbf{u}}{|\mathbf{v}^* \mathbf{u}|}$ 
     $z_{k_{\text{predict}}} = z_{k-1} + \tau r_k$ 
    ÉTAPE 2 : Correction par une itération de la méthode de Newton
    calcul de  $(s_{\min}, u_{\min}, v_{\min})$  pour  $z_{k_{\text{predict}}}$   $I - A$ 
    calcul de  $z_k$  selon l'équation  $z_k = z_{k_{\text{predict}}} - \frac{(s_{\min} - \varepsilon)}{(\mathbf{u}^* \mathbf{v})}$ .
  fin tant que
fin pour

```

---

## 4.4 Exemple

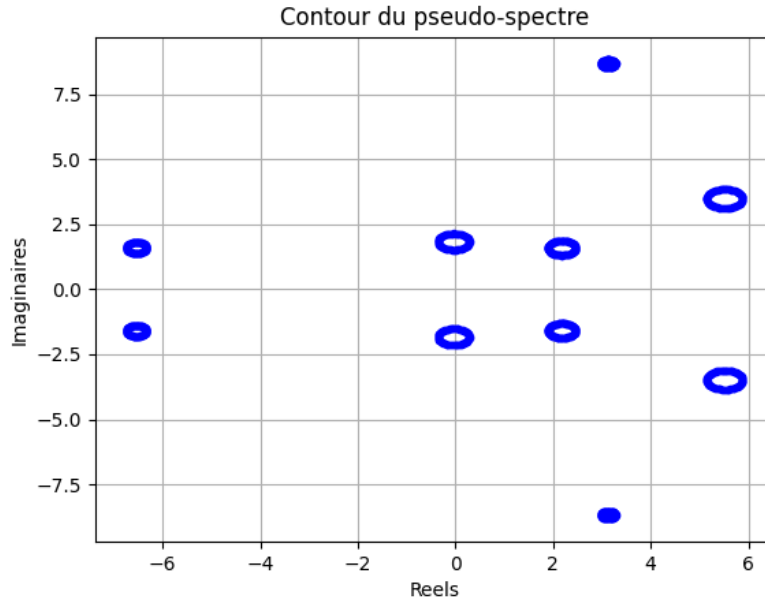


FIGURE 4 – Pseudospectre tracé par Prédiction-correction pour la même matrice,  $\varepsilon = 0.1, tol_{\text{contour}} = 0.01$

## 4.5 Analyse de l'algorithme

On trouve une dépendance entre la tolérance pour terminer le calcul d'une composante et  $\varepsilon$ . En effet, si on diminue  $\varepsilon$  en fixant la tolérance, la forme obtenue devienne imprécise. Il faut donc diminuer la tolérance de contour, sauf que cela tout seul, donne des problèmes de convergence. Il nous faut donc aussi diminuer le pas pour chaque prédiction, et dans ce cas là, on a pas de problèmes : on peut diminuer  $\varepsilon$  et augmenter  $n$  autant que l'on veut.

## 5 Comparaison de performances

Pour confirmer nos résultats et comparer les performances de GRID et de la méthode de prédiction-correction, on a dessiné les graphes du temps d'exécutions de ces fonctions en fonction de la taille  $n$  de la matrice (qu'on a fait varier de 5 à 35). Comme attendu, GRID qui nécessite beaucoup plus de calculs de valeurs singulières est également beaucoup plus lent.

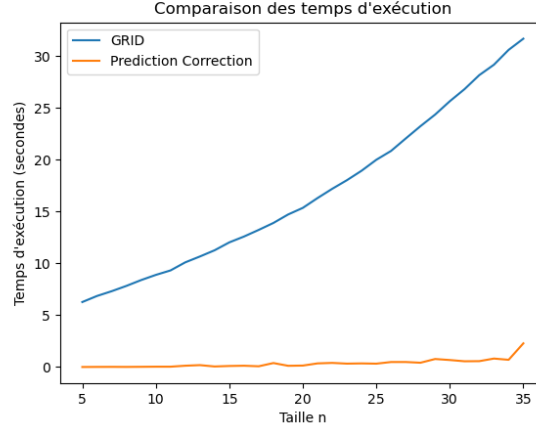


FIGURE 5 – Comparaison des performances de GRID (en bleu) et prédiction-correction (en orange)

## 6 Abscisse et rayon du pseudospectre

### 6.1 Motivation

On a vu précédemment deux manières de calculer intégralement le pseudospectre. En réalité, ce n'est pas forcément ce que l'on souhaite, car on a rarement besoin de connaître l'entière du pseudospectre. On va plutôt préférer certaines caractéristiques de celui-ci. En l'occurrence, on a surtout besoin, avec  $A \in M_n(\mathbb{C})$ , de :

$$\alpha_\varepsilon(\mathbf{A}) = \sup_{z \in \sigma_\varepsilon(\mathbf{A})} \operatorname{Re} z, \quad \rho_\varepsilon(\mathbf{A}) = \sup_{z \in \sigma_\varepsilon(\mathbf{A})} |z|,$$

où  $\alpha_\varepsilon(\mathbf{A})$  est appelé abscisse du pseudospectre et  $\rho_\varepsilon(\mathbf{A})$  est appelé rayon du pseudospectre. En d'autres termes, ce sont respectivement le point du pseudospectre avec la plus grande partie réelle et le point de plus grande norme qui se situent sur la bordure de  $\sigma_\varepsilon$  notée  $\partial\sigma_\varepsilon$ . Les algorithmes utilisés pour étudier la stabilité des systèmes sont à l'origine des algorithmes dont nous allons nous servir pour calculer  $\alpha_\varepsilon(\mathbf{A})$  et  $\rho_\varepsilon(\mathbf{A})$ . On pourrait simplement approximer le pseudospectre grâce à la méthode de prédiction-correction puis y trouver naïvement  $\alpha_\varepsilon(\mathbf{A})$  et  $\rho_\varepsilon(\mathbf{A})$  mais ça serait à la fois imprécis et peu efficace, comme ça nécessiterait beaucoup de calculs que l'on peut éviter comme on va le voir. Etudions le calcul de l'abscisse du pseudospectre dont l'algorithme est plus accessible. Celui du rayon est, bien que similaire, plus compliqué.

### 6.2 Abscisse du pseudospectre : algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$

Soit  $A \in M_n(\mathbb{C})$ .

Pour pouvoir utiliser cet algorithme, on aura besoin de savoir trouver l'intersection de  $\partial\sigma_\varepsilon(A)$  avec des droites verticales et horizontales. Pour se faire, on a les résultats ci-dessous.

**Théorème 6.1** ([TE05] 42.1). *La matrice  $(x + iy) - \mathbf{A}$  a pour valeur singulière  $\varepsilon$  si et seulement si  $iy$  est valeur propre de*

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

**Théorème 6.2** ([TE05] 42.2). *La matrice  $(x + iy) - \mathbf{A}$  a pour valeur singulière  $\varepsilon$  si et seulement si  $ix$  est valeur propre de*

$$\begin{pmatrix} -y + i\mathbf{A}^* & -\varepsilon \mathbf{I} \\ \varepsilon \mathbf{I} & i\mathbf{A} + y \end{pmatrix}.$$

Avec ces outils à disposition, expliquons l'algorithme Criss-cross permettant de calculer  $\alpha_\varepsilon(\mathbf{A})$ . D'abord, on calcule la valeur propre la plus à droite (i.e de plus grande partie réelle) de  $\mathbf{A}$  que l'on note  $\lambda$ . On trouve le point  $z_1$  le plus à droite de  $\partial\sigma_\varepsilon(A)$  qui intersecte la droite horizontale  $\{z \in \mathbb{C} : \operatorname{Im} z = \operatorname{Im} \lambda\}$  passant par  $\lambda$ . C'est donc une borne inférieure pour  $\alpha_\varepsilon(\mathbf{A})$ .

Pour  $k$  entier naturel non nul, jusqu'à convergence : on trouve tous les points auxquels  $\partial\sigma_\varepsilon(A)$  intersecte la droite verticale  $\{z \in \mathbb{C} : \operatorname{Re} z = \operatorname{Re} z_k\}$  passant par les  $z_k$ . On obtient donc des intervalles sur cette droite. On calcule les milieux de ces intervalles qui appartiennent à l'adhérence de  $\sigma_\varepsilon(A)$  puis à partir de ces points milieux on fait une recherche horizontale pour trouver l'intersection la plus à droite avec  $\partial\sigma_\varepsilon(A)$ . De toutes les nouvelles intersections obtenues, on ne conserve que celle la plus à droite et on la note  $z_{k+1}$ . On itère ce processus autant que nécessaire.

Donnons plus de détails sur la manière dont on trouve les intervalles mentionnés précédemment.

On a que  $(x + iy) - \mathbf{A}$  a pour valeur singulière  $\varepsilon$  si et seulement si  $iy$  est valeur propre de la première

matrice augmentée mais on a  $x + iy \in \partial\sigma_\varepsilon(A)$  si et seulement si  $\varepsilon$  est valeur singulière minimale. Ainsi, pour chacune des valeurs propres purement imaginaires de la matrice augmentée, il faut calculer la plus petite valeur singulière de  $(x + iy) - \mathbf{A}$ .

Ce calcul n'est pas requis pour la recherche horizontale car on a que pour la plus grande des valeurs propres purement imaginaires de la matrice augmentée,  $\varepsilon$  est obligatoirement valeur singulière minimale.

### 6.3 Illustration de l'algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$

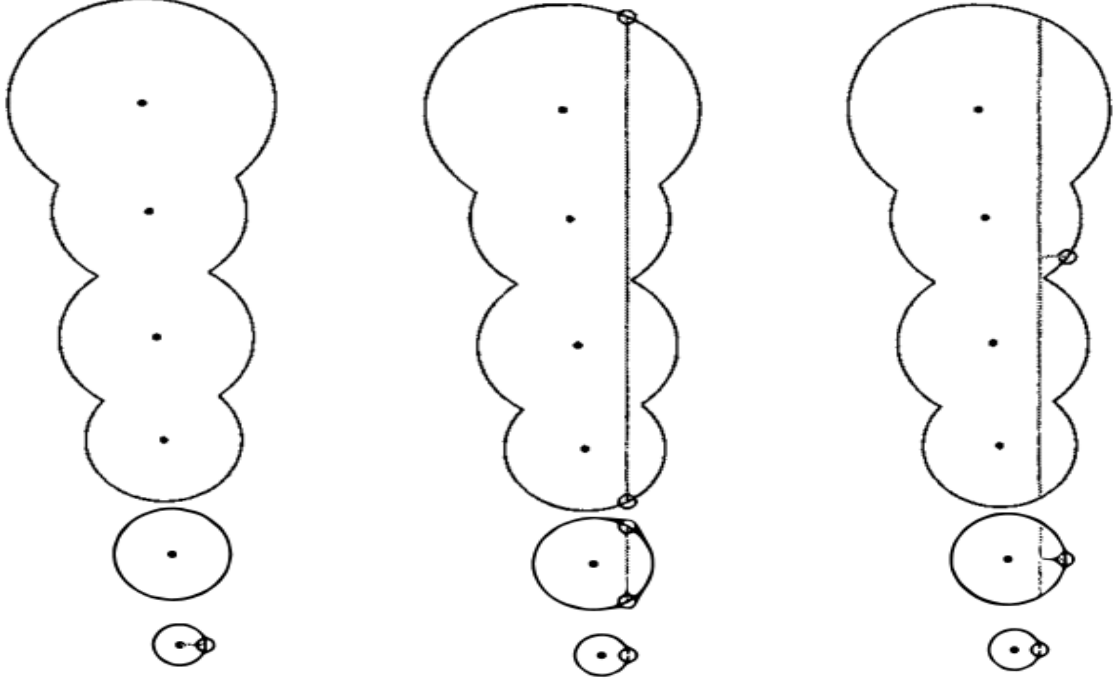


FIGURE 6 – représentation d'une itération de l'algorithme Criss-cross avec le calcul de la valeur propre la plus à droite puis une recherche horizontale (à gauche), puis le calcul des intersections verticales (au milieu) puis, enfin, le calcul des intersections horizontales à partir des points du milieu des intervalles trouvés (à droite)

### 6.4 Pseudo-code de l'algorithme Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$

Soit  $A \in M_n(\mathbb{C})$ ,  $\lambda_{\max}$  la valeur propre la plus à droite de  $A$ , *Midpoints* un tableau contenant les points milieux (initialisé à  $\text{Im } \lambda_{\max}$ ),  $X$  et  $Y$  des tableaux contenant les parties réelles et imaginaires (i.e les coordonnées) des points étudiés.

---

#### Algorithme 3 Criss-cross pour $\alpha_\varepsilon(\mathbf{A})$

---

**tant que**  $(|X - \text{Midpoint}| > \text{tol})$  **faire**

    ÉTAPE 1 : recherche horizontale à partir de  $\lambda_{\max}$

    mettre dans  $X$  la plus grande valeur propre purement imaginaire de  $\begin{pmatrix} -y + i\mathbf{A}^* & -\varepsilon\mathbf{I} \\ \varepsilon\mathbf{I} & i\mathbf{A} + y \end{pmatrix}$

    mettre à jour le point  $z$  de coordonnées  $(x, y)$  le plus à droite de  $\partial\sigma_\varepsilon$

    ÉTAPE 2 : recherche verticale à partir de  $z$

    mettre dans  $Y$  les valeurs propres purement imaginaires de  $\begin{pmatrix} x - \mathbf{A}^* & -\varepsilon\mathbf{I} \\ \varepsilon\mathbf{I} & \mathbf{A} - x \end{pmatrix}$

$k = 0$

**tant que**  $2k + 1 < \text{len}(Y)$  **faire**

        ajouter à *Midpoints*  $\frac{Y[2k] + Y[2k+1]}{2}$

**fin tant que**

**fin tant que**

retourner  $X[0]$  qui vaut  $\alpha_\varepsilon(\mathbf{A})$

---



## 7 Pseudospectre par composantes

### 7.1 Introduction

On rappelle que pour  $\|\cdot\| = \|\cdot\|_2$ , la distance d'une matrice  $A \in M_n(\mathbb{C})$  à l'ensemble des matrices singulières

$$\sigma(A) = \min\{\varepsilon \geq 0 : \exists \Delta \in \mathbb{C}^{n \times n} \text{ telle que } \|\Delta\| \leq \varepsilon \text{ et } A + \Delta \text{ est singulière}\},$$

vaut  $s_{\min}(A)$ .

Dans le but de traiter le cas de perturbations structurées, la notion de rayon de non-singularité a été introduite :

$$\sigma(A, B, C) = \min\{\varepsilon \geq 0 : \exists \Delta \in \mathbb{C}^{n \times n} \text{ telle que } \|\Delta\| \leq \varepsilon \text{ et } A + C\Delta B \text{ est singulière}\},$$

où  $B$  et  $C$  sont des matrices décrivant les patterns de perturbations et retranscrivent ainsi des informations quant aux perturbations.

Dans le contexte de perturbations par composantes,  $\sigma(A)$  et  $\sigma(A, B, C)$  sont remplacés par

$$\sigma(A, E) = \min\{\varepsilon \geq 0 : \exists \tilde{E} \in \mathbb{C}^{n \times n} \text{ telle que } |\tilde{E}| \leq \varepsilon E \text{ et } A + \tilde{E} \text{ est singulière}\},$$

où  $E$  est une matrice positive. Pour arriver à des perturbations par composantes, on décide dans ce projet d'utiliser  $E = |A|$ . Ainsi seuls les éléments non nuls de  $A$  sont soumis à des perturbations.

Les inégalités et valeurs absolues utilisées relativement aux matrices sont ainsi appliquées élément par élément.

Pour  $A \in \mathbb{R}^{n \times n}$  et  $E$  la matrice Attila (n'ayant que des uns), le calcul de  $\sigma(A, E)$  est NP-difficile ce qui se retranscrit par : pour  $A \in \mathbb{Q}^{n \times n}$  et  $r \in \mathbb{Q}$ , la preuve de l'inégalité qui suit est NP-difficile

$$\min\{\varepsilon \geq 0 : \exists \tilde{E} \in \mathbb{R}^{n \times n} \text{ telle que } |\tilde{E}| \leq \varepsilon E \text{ et } A + \tilde{E} \text{ est singulière}\} \leq r,$$

En revanche on a pour  $A$  une matrice non singulière

$$\frac{1}{\rho(|A^{-1}E|)} \leq \sigma(A, E) \leq \frac{(3 + 2\sqrt{2})n}{\rho(|A^{-1}E|)},$$

où  $\rho(\cdot)$  est le rayon spectral i.e la plus grande valeur propre en module.

La définition précédente de  $\sigma(A, E)$  et la dernière inégalité permettent d'introduire une nouvelle notion, le  $\varepsilon$ -pseudospectre par composantes, défini comme suit :

**Définition 5** ([MS04]).

$$\Lambda_\varepsilon(A, E) = \{z \in \mathbb{C} : \exists \Delta \in \mathbb{C}^{n \times n} \text{ telle que } |\Delta| \leq \varepsilon E \text{ et } A + \Delta - zI \text{ est singulière}\},$$

qui est également égal à

$$\{z \in \mathbb{C} : \sigma(A - zI, E) \leq \varepsilon\}.$$

On note que la première définition du  $\varepsilon$ -pseudospectre par composante est simplement l'adaptation de la définition du  $\varepsilon$ -pseudospectre  $\sigma_\varepsilon(A) = \{z \in \mathbb{C} : \exists \Delta \in \mathbb{C}^{n \times n} \text{ telle que } \|\Delta\|_2 \leq \varepsilon \text{ et } A + \Delta - zI \text{ est singulière}\}$  et la seconde est l'adaptation de  $\sigma_\varepsilon(A) = \{z \in \mathbb{C} : \sigma_{\min}(A - zI) \leq \varepsilon\}$ .

On a alors immédiatement les inclusions :

$$\Gamma_{\varepsilon/(3+2\sqrt{2})n}(A, E) \subseteq \Lambda_\varepsilon(A, E) \subseteq \Gamma_\varepsilon(A, E),$$

où

$$\Gamma_\varepsilon(A, E) = \left\{ z \in \mathbb{C} : \rho(|(A - zI)^{-1}| E) \geq \frac{1}{\varepsilon} \right\}.$$

On peut donc utiliser comme pseudospectre par composantes les contours de  $z \in \mathbb{C} \mapsto f(z) = \rho(|(A - zI)^{-1}| E)$ .

### 7.2 Pseudo-code du calcul du pseudospectre par composantes

Soit  $A \in M_n(\mathbb{C})$ . On pose  $E = |A|$ . On note  $m$  le nombre de points  $z$  en lesquels on calcule  $f(z) = \rho(|(A - zI)^{-1}| E)$ , points dont les coordonnées  $(x, y)$  sont inscrits dans des tableaux  $X$  et  $Y$ . Ces  $m$  points sont pris aléatoirement dans  $n$  les disques de Gershgorin adaptés. On note  $F$  le tableau dans lequel figurent les valeurs  $f(z)$  calculées. On trace le contour de  $F$  (selon la courbe de niveau  $C = \frac{1}{\varepsilon}$ ) et on obtient un encadrement de  $\Lambda_\varepsilon(A, E)$ . On suit le pseudo-code suivant, similaire à celui de GRID.

---

**Algorithme 4** Calcul du pseudospectre par composantes

---

```
pour  $i = 0$  à  $n - 1$  faire
  pour  $j = 0$  à  $m - 1$  faire
    pour  $k = 0$  à  $m - 1$  faire
       $F[j, k] = f(A, n, X[j, k], Y[j, k], E)$ 
    fin pour
  fin pour
  contour( $X, Y, F, \frac{1}{\varepsilon}$ )
fin pour
```

---

### 7.3 Exemple

Ci-dessous les résultats obtenus pour une matrice de taille  $n = 10, \varepsilon = 0.1$  et 500 points.

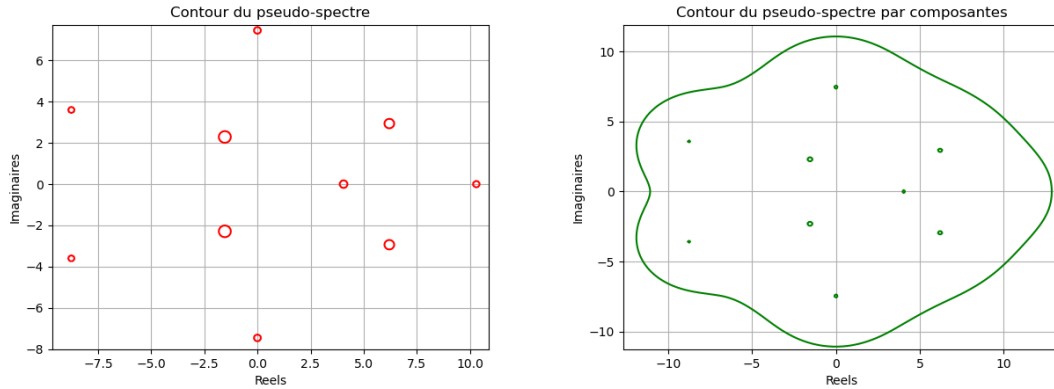


FIGURE 7 – Pseudospectre tracé par GRID (à gauche) et pseudospectre par composantes (à droite)

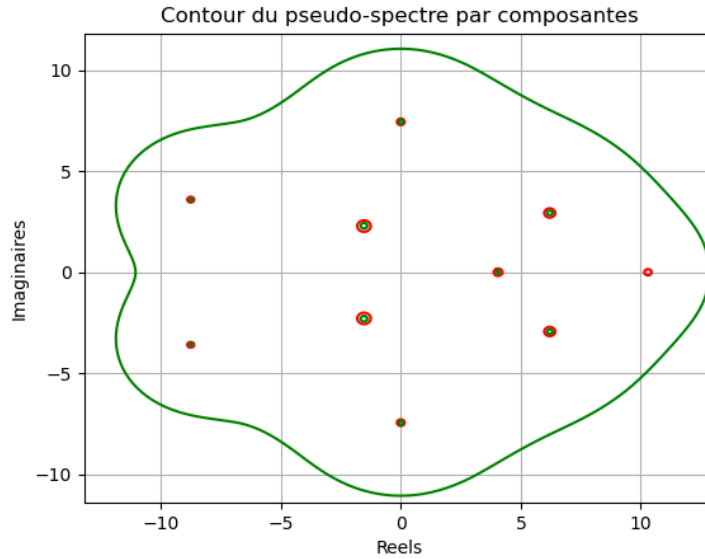


FIGURE 8 – Pseudospectre par composantes (en vert), pseudospectre (en rouge)

## 8 Conclusion

On a donc pu introduire la notion de pseudospectre, étudier des algorithmes plus (prédiction-corrrection) ou moins (GRID) efficaces permettant de le calculer et étendre sur une approche de perturbation par composantes, menant à l'étude du pseudospectre par composantes et un algorithme permettant de le calculer. On a également pu mettre en place l'algorithme Criss-cross permettant d'obtenir des informations quant aux propriétés du pseudospectre (son abscisse et son rayon), bien que l'étude du rayon dont le code est en partie écrit n'ait pas abouti. Tout le code relatif à ce projet est disponible en libre accès [ici](#).

## Références

- [Brü96] Martin BRÜHL. « A curve tracing algorithm for computing the pseudospectrum ». In : *BIT Numerical Mathematics* 36.3 (sept. 1996), p. 441-454. ISSN : 1572-9125. DOI : [10.1007/BF01731926](https://doi.org/10.1007/BF01731926). URL : <https://doi.org/10.1007/BF01731926>.
- [MS04] A.N MALYSHEV et M SADKANE. « Componentwise pseudospectrum of a matrix ». In : *Linear Algebra and its Applications* 378 (2004), p. 283-288. ISSN : 0024-3795. DOI : <https://doi.org/10.1016/j.laa.2003.10.005>. URL : <https://www.sciencedirect.com/science/article/pii/S0024379503008024>.
- [TE05] Lloyd N. TREFETHEN et Mark EMBREE. *The Behavior of Nonnormal Matrices and Operators*. Princeton : Princeton University Press, 2005. ISBN : 9780691213101. DOI : [doi:10.1515/9780691213101](https://doi.org/10.1515/9780691213101). URL : <https://doi.org/10.1515/9780691213101>.