Sorbonne Université

UE 4IN900 - « PROJET », M1 Speciality SFPN

Project Report
COMPUTATION OF PSEUDOSPECTRUM

Calle Viera Andersson, Zaghouani Slim

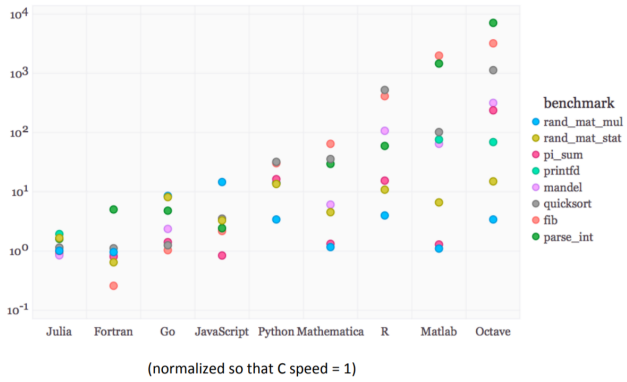
Supervised by
GRAILLAT Stef

Contents

Introduction

Our research subject is the computation of Pseudospectrum. A Pseudospectra is a mathematical tool used to describe the behavior of a linear transformation when its eigenvalue analysis is misleading and when predictions fail to match observations. Specifically, trouble may arise when the associated sets of eigenvectors are ill-conditioned with respect to the norm, here we will focus mainly on the case of the familiar Euclidean or 2-norm. This issue frequently arises in several fields, from atmospheric science to non-Hermitian quantum mechanics, so it is crucial to find a quick way to compute the Pseudospectra of a matrix. During this project we will study the mathematical properties of Pseudospectrums, eigenvalues and Singular Value Decomposition and compare the performances of the sequential and parallel implementations of two algorithms (GRID and prediction-correction).

To implement the algorithms presented through the paper, we will use the **Julia** programming language and more specifically **Julia Pro-1.2.0-2**. Created in 2009, Julia is a compiled language, such as C, and not interpreted for faster runtime performance but is also interactive like Python and offers a pretty much straightforward syntax very much like Python or Matlab. Not only Julia combines the benefits of this languages but it can also call Python, C, and Fortran libraries. One other Julia's most distinctive features is that it supports parallel computing(with or without MPI), which makes it well suited for high performance numerical analysis. Finally it offers a great and simple collection of methods to create complex U.Is.



Performance of different languages (normalized so that C speed = 1).

#Still working on it but :
Brief description of our functions and how to launch/ use them.

1 Pseudo-spectra of a matrix

Definitions :

Before starting with the core of the project let's remind the definition of a matrix norm. In what follows, $\mathbb{K}$ will denote a field.
Let $A, B \in M_n(\mathbb{K})$ and $\alpha \in \mathbb{K}$, the function $\|\cdot\| : M_n(\mathbb{K}) \longrightarrow \mathbb{R}$ is a matrix norm if it satisfies the following properties:

1. $\|A\| \geq 0$ and $\|A\| = 0 \Leftrightarrow A = 0$
2. $\|\alpha A\| = |\alpha|\|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$
4. $\|AB\| \leq \|A\|\|B\|$

In this project we will be using the 2-norm definided as :

$$\|A\|_2 = \max_{\|x\|\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \tag{1}$$

It exists various equivalent definitions of pseudospectra given and demonstrated in [1] for example. Here we will use the following ones :

Definition 1.1. Let $A \in M_n(\mathbb{C})$ and $\varepsilon > 0$.
The ε-pseudospectrum $\Lambda_\varepsilon(A)$ of A is the set of $z \in \mathbb{C}$ such that :

$$\|(z - A)^{-1}\|_2 > \varepsilon^{-1} \tag{2}$$

Definition 1.2. $\Lambda_\varepsilon(A)$ is the set of $z \in \mathbb{C}$ such that for some $E \in M_n(\mathbb{C})$ with $\|A - E\|_2 \leq \varepsilon$.

$$z \in \Lambda(E) \tag{3}$$

Definition 1.3. $\Lambda_\varepsilon(A)$ is the set of $z \in \mathbb{C}$ such that for some $v \in \mathbb{C}^n$ with $\|v\| = 1$:

$$\|(z - A)v\|_2 < \varepsilon \tag{4}$$

Let E $\in M_n(\mathbb{C})$, to fully grasp the next definition let's remind the reader that we call singular vlaues of E the square roots of the matrix's eigenvalues. We will write σ_{min} the lowest singular value.

Definition 1.4. For $\|\cdot\| = \|\cdot\|_2$, $\Lambda_\varepsilon(A)$ is the set of $z \in \mathbb{C}$ such that :

$$\sigma_{min}(z - A) \leq \varepsilon \tag{5}$$

Theorem 1.1. For any matrix $A \in M_n(\mathbb{C})$ and for $\|\cdot\|_2$ the four definitions above are equivalent.

Gershgorin discs :

Definition 1.5. Let $A = (a_{ij}) \in M_n(\mathbb{C})$.
For any $i \in [\![1,\ n]\!]$, we call Gershgorin discs associated to A :

$$D_i = B_f\left(a_{ii},\ \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\right) = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \right\} \tag{6}$$

Theorem 1.2 (Gerschgorin theorem)**.**
Let $A = (a_{ij}) \in M_n(\mathbb{C})$, every eigenvalue of A lies within at least one of the Gershgorin discs
i.e if λ is an eigenvalue of A then $\lambda \in \bigcup_{i=1}^{n} D_i$.

2 GRID-svd algorithme

All the algorithms presented here are to compute the pseudospectra of matrix are based on the formula in (1.4) and. The easiest way, but not the most efficient one, to determine the pseudospectra of A is to use the Basic GRID SVD algorithm which consists of checking if a previously defined set of points on a grid satisfy the condition of (5) and ploting the ones which do.
To determine this grid of points let's use the Theorem 1.2 and extend it to the pseudospectra of A.

Theorem 2.1. Let $A = (a_{ij}) \in M_n(\mathbb{C})$ and $\varepsilon > 0$. For any $z \in \Lambda_\varepsilon(A)$:

$$|z - a_{ii}| \leq \sqrt{n}\varepsilon + \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \tag{7}$$

Proof. Let z $\in \Lambda_\varepsilon(A)$, and $B \in M_n(\mathbb{C})$ such that $\|B - A\|_2 \leq \varepsilon$ as in Definition 1.2.
Let $\|\cdot\|_\infty$ denote the sup norm defined such as $\|A\|_\infty = \max\limits_{0\leq i\leq n} \sum\limits_{j=1}^{n} |a_{ij}|$, using the fact that on a finite-dimensional linear space two norms are equivalent, we have here :

$$\frac{1}{\sqrt{n}}\|A - B\|_2 \leq \|A - B\|_\infty \leq \sqrt{n}\|A - B\|_2 \tag{8}$$

$$\implies \|A - B\|_\infty = \max_{0\leq i\leq n} \sum_{j=1}^{n} |a_{ij} - b_{ij}| \leq \sqrt{n}\|A - B\|_2 \leq \sqrt{n}\varepsilon$$

$\forall i \geq 1$, $|z - a_{ii}| = |z - b_{ii} + b_{ii} - a_{ii}| \leq |z - b_{ii}| + |b_{ii} - a_{ii}|$
as, $|b_{ii} - a_{ii}| \leq \sum\limits_{j=1}^{n} |b_{ij} - aij| \leq \max\limits_{0\leq i\leq n} \sum\limits_{j=1}^{n} |a_{ij} - b_{ij}| \leq \sqrt{n}\varepsilon$ and, $|z - b_{ii}| \leq \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}|$ by Definition 1.5.

$$\implies |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}| + \sqrt{n}\varepsilon$$

Using the reverse triangle inequality : $\forall i, j \geq 1$, $|a_{ij} - b_{ij}| \geq \big||a_{ij}| - |b_{ij}|\big|$ we can observe that

$$\sqrt{n}\varepsilon \geq \max_{0\leq i\leq n} \sum_{j=1}^{n} |a_{ij} - b_{ij}| \geq \sum_{j=1}^{n} |a_{ij} - b_{ij}| \geq \sum_{j=1}^{n} \big||a_{ij}| - |b_{ij}|\big| \geq \Big|\sum_{j=1}^{n} |a_{ij}| - |b_{ij}|\Big|$$

$$\implies -\sqrt{n}\varepsilon \leq \sum_{j=1}^{n} \big(|a_{ij}| - |b_{ij}|\big) \leq \sqrt{n}\varepsilon \implies \sum_{j=1}^{n} |b_{ij}| \leq \sum_{j=1}^{n} |a_{ij}| + \sqrt{n}\varepsilon$$

$$\iff \Big(\sum_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}|\Big) + |b_{ii}| \leq \Big(\sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\Big) + |a_{ii}| + \sqrt{n}\varepsilon \iff \sum_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| + \sqrt{n}\varepsilon - (|b_{ii}| - |a_{ii}|)$$

Note that $|b_{ii}| - |a_{ii}| \leq \big||b_{ii}| - |a_{ii}|\big| \leq |a_{ij} - b_{ij}| \leq \sqrt{n}\varepsilon$
Hence : $\sum\limits_{\substack{j=1 \\ j\neq i}}^{n} |b_{ij}| \leq \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| + \sqrt{n}\varepsilon - \sqrt{n}\varepsilon = \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$

$$\implies |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| + \sqrt{n}\varepsilon$$

□

We can now use this formula to determinate a domain of research for the pseudo-eigenvalues of a matrix and in doing so optimize the time of execution of our algorithm.
One could argue that depending on the localization and density of the pseudospectra, an algorithm based on this ineqality could spend time calculating non relevant values but it is a great first step.

Algorithms:

Gershgorin discs :

For the rest of the section assume that we have $A \in M_n(\mathbb{C})$, to begin with we used the formula in (7) to compute n discs that we stored as a list of centers and rays to be used later on.

Algorithm 1: `cerclesG`

Data: $A \in M_n(\mathbb{C})$, $\varepsilon \in \mathbb{R}$
Result: $Disk = ((c_1,\ r_1), \dots, (c_n,\ r_n))$ a list of centers and rays where
$\forall i \in [\![1,\ n]\!],\ c_i \in \mathbb{C}, r_i \in \mathbb{R}_+$
Initialization :
$Disk = [\,]$
$n = size(A)$
$k = \sqrt{n}\varepsilon$
for $i = 1$ **to** n **do**
  $summ = 0$
  for $j = 1$ **to** n **do**
    if $i \neq j$ **then**
      $summ = summ + |A[i, j]|$
  $A[i] = (A[i,\ i],\ summ + k)$
return *Disk*

Once those discs are computed we can calculate the extreme down right corner of the smallest square containing all our discs. Once this point is found we then return and save it's coordinates and also the lenght of the square. This allows us to store less data and makes it easier for the rest of the algorithm. Now that we have limited our search to a much small portion of space we can make a grid of it and use the inequality in (5) to test if it is, indeed, in the pseudospectra. This is the method that was used in 1991 and can be optimized now. Indeed `grid_svd` is not optimal at all because in order to determine wether or not a point of the grid is in the pseudospectra we have to compute a full SVD. Let $m^2 \in \mathbb{N}$ be the size of the grid, we therefore have a complexity of $\mathcal{O}(n^3m^2)$.

Algorithm 2: `grid_svd`

Data: $A \in M_n(\mathbb{C})$, $\varepsilon \in \mathbb{R}$, $nb_{points} \in \mathbb{N}$
Result: $p : plot$
Initialization :
$eig = \texttt{eigvals}(A)$
$circles = \texttt{cerclesG}(A,\ \varepsilon)$
$contour = \texttt{build\_contour}(circles)$
$x = \texttt{linspace}(contour[1],\ contour[1] + contour[3],\ nb_{points})$
$y = \texttt{linspace}(contour[2],\ contour[2] + contour[4],\ nb_{points})$
$Z = [\,]$
for $K = 1$ **to** nb_{points} **do**
  for $j = 1$ **to** nb_{points} **do**
    $Z[j,\ k] = \texttt{min}(\texttt{svd}(\texttt{diag}(x[k] + \mathbf{i}y[j]) - A))$
$p = \texttt{contour}(x,\ y,\ Z,\ levels = \varepsilon)$
$p = \texttt{scatter}(\mathrm{Re}(eigs),\ \mathrm{Im}(eigs))$
return p

Results:

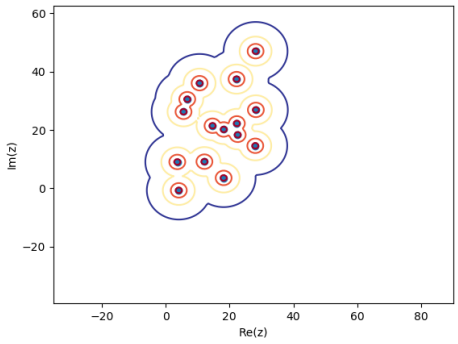


Figure 2: ε-pseudospectre computed with our function for A

We first tested our function for a small random diagonal matrix $A \in \mathbb{C}^{15\times 15}$ and for $\varepsilon \in \{1,\ 2.5,\ 5, 10\}$ and with 200×200 points on the grid. The time of computation is about `1sec` with `1.32 M` allocations, `1.168 GiB` and `21.11% gc time`.

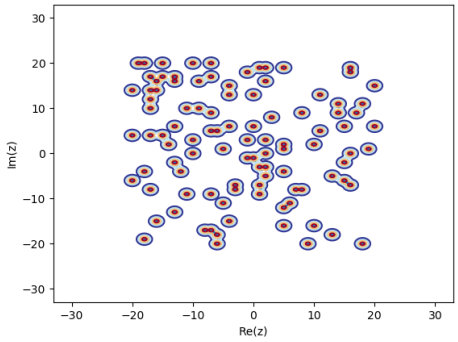


Figure 3: ε-pseudospectre computed with our function for B

Then we tested our function for a bigger random diagonal matrix $B \in \mathbb{C}^{100\times 100}$ and $\varepsilon \in \{0.4,\ 0.7,\ 1, 1.3\}$ and with 200×200 points on the grid. The time of computation is about `23.033569sec` with `1.03 M` allocations, `22.422 GiB` and `15.89% gc time`.

3 Prediction-correction

Another view on the ε-pseudospectrum $\Lambda_\varepsilon(A)$ is that it can also be obtained by tracing directly its boundary curve. By using def. (1.1) it is the set $\{z \in \mathbb{C} : \|(z-A)^{-1}\| = \varepsilon^{-1}\}$ or equivalently by using def. (1.4), $\{z \in \mathbb{C} : \sigma_{min}(z-A) = \varepsilon\}$. In order to obtain this boundary curve we use a predictor-corrector path following algorithm which works like this :

1. We determine z_1 on $\partial\Lambda_\varepsilon(A)$ ($i.e$ it is on the boundary curve)
2. for $k \geq 2$:
   (a) (Prediction) : determine a direction $r_k \in \mathbb{C}$, $|r_k| = 1$ and a steplength τ_k, then compute the point $\widetilde{z}_k = z_{k-1} + \tau_k r_k$.
   (b) (Correction) : determine a direction $d_k \in \mathbb{C}$, $|d_k| = 1$ and θ_k, then compute the corrected point $z_k = \widetilde{z}_k + \theta_k d_k$.

Algorithms:

4 Componentwise GRID

The main idea for this algorithm is to take into consideration a perturbation for each coefficient of the matrix. Given a weights matrix E we will define the componentwise pseudospectrum of a matrix such as a contour sets of a function.

Algorithms:

Algorithm 3: `grid_par_composante`

Data: $A \in M_n(\mathbb{C})$, $\varepsilon \in \mathbb{R}$, $nb_{points} \in \mathbb{N}$
Result: $p : plot$
Initialization :
$eig = \texttt{eigvals}(A)$
$circles = \texttt{cerclesG}(A,\ \varepsilon)$
$contour = \texttt{build\_contour}(circles)$
$x = \texttt{linspace}(contour[1],\ contour[1] + contour[3],\ nb_{points})$
$y = \texttt{linspace}(contour[2],\ contour[2] + contour[4],\ nb_{points})$
$E = (1)_{n \times n}$
$Z = [\,]$
for $K = 1$ **to** nb_{points} **do**
  for $j = 1$ **to** nb_{points} **do**
    $M = |(\texttt{diag}(x[k] + \mathbf{i}y[j]) - A)^{-1}|$
    $N = ME$
    $Z[j,\ k] = \texttt{max}(\texttt{eigvals}(N))$
$p = \texttt{contour}(x,\ y,\ Z,\ levels = \varepsilon)$
$p = \texttt{scatter}(\mathrm{Re}(eigs),\ \mathrm{Im}(eigs))$
return p

Results:

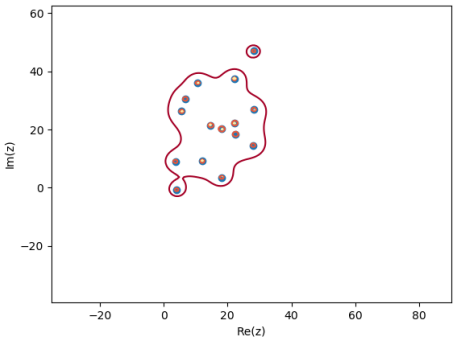


Figure 4: ε-pseudospectre computed with our function for B

For our first matrix A the time of computation is about `3.552562sec` with `1.23 M` allocations, `2.030 GiB` and `16.45% gc time`.

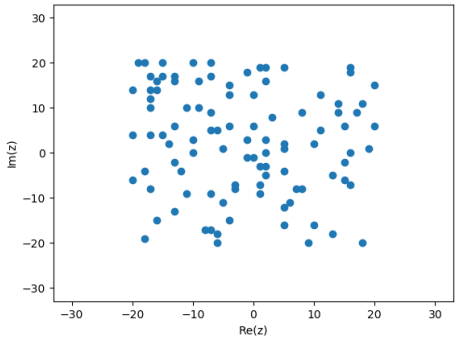


Figure 5: ε-pseudospectre computed with our function for B

The second matrix $B$with the same parameters takes a lot of time but is at the same time more precise. The time of computation is about `152.277965sec` with `1.39 M` allocations, `28.564 GiB` and `3.06% gc time`.

Conclusion

Still working on it ...