

Lab 1

Pseudospectra

Lab Objective: *The Pseudospectrum of a matrix is related to its eigenvalues. The Pseudospectra of normal and non-normal matrices will be investigated. We will plot pseudospectra and examine the characteristics of different types of matrices.*

Pseudospectra Definition

Eigenvalues of matrices are extremely important, as there are a myriad of practical applications in which they are used. In Quantum Mechanical systems, particularly, many operators can be represented by matrices, and the eigenvalues are the physical observables of the system. Though we can make extremely precise measurements, there is often an ϵ -small error in measurements. In this case, we are interested not just in the eigenvalues, but in the “almost” eigenvalues, the ϵ region surrounding the actual eigenvalue. The ϵ region around the eigenvalues of a square matrix A represents the ϵ -pseudospectrum (plural *pseudospectra*) of A , which can be denoted $\sigma_\epsilon(A)$.

To be more precise, the *point spectrum* of A is defined as

$$\begin{aligned}\sigma(A) &= \{z \in \mathbb{C} : \det(zI - A) = 0\} \\ &= \{z \in \mathbb{C} : (zI - A)^{-1} \text{ is undefined}\}.\end{aligned}$$

The point spectrum of A contains all the eigenvalues of A . By convention, if z is an eigenvalue of A , then $\|(zI - A)^{-1}\| = \infty$. The ϵ -pseudospectrum of a matrix is a generalization of its point spectrum, and for a matrix A is defined as

$$\sigma_\epsilon(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\| > \epsilon^{-1}\}. \quad (1.1)$$

Note that this definition requires the use of a matrix norm. Any matrix norm can be used, but in this lab we will use the norm induced by the 2-norm.

In the case of the 2-norm, the pseudospectrum has an equivalent definition:

$$\sigma_\epsilon(A) = \{z \in \mathbb{C} : z \in \sigma(A + E) \text{ for some } E \text{ with } \|E\| < \epsilon\}. \quad (1.2)$$

When the entries of a matrix are slightly altered, changes in the point spectrum also occur. The pseudospectrum can then be thought of as the set of eigenvalues that

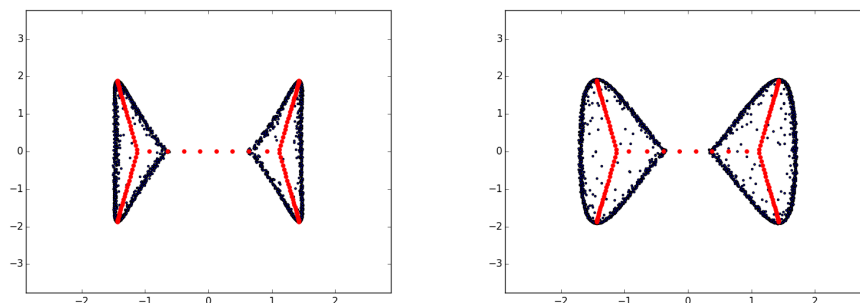


Figure 1.1: Scatter plots representing the ϵ -pseudospectrum of the matrix given in Problem 1. The plot on the left uses $\epsilon = 10^{-6}$. The plot on the right uses $\epsilon = 10^{-3}$. In both plots, the eigenvalues of twenty matrices $A + E$ were plotted.

can result from small changes in the entries of a matrix. If A is an $n \times n$ matrix, then we can use Equation 1.2 with some $n \times n$ matrix E satisfying $\|E\| < \epsilon$ to find the ϵ -pseudospectrum of A , which will be a set of all complex numbers z which are in the point spectrum of $A + E$.

One means of plotting the ϵ -pseudospectrum of A is by generating several random matrices E with norm ϵ , and then plotting the eigenvalues z of all of them together. The horizontal axis is the real part of z , with the vertical axis being the imaginary part of z .

Problem 1. Complete the function below for plotting the pseudospectrum of a matrix A by plotting the point spectra of several matrices $A + E$, where $\|E\| = \epsilon$.

```
def ps_scatter_plot(A, epsilon=.001, num_pts=20):
    '''Plots the 'poorman's pseudospectrum' of a matrix A

    Parameters:
    A : ndarray of size (n,n)
        The matrix whose pseudospectrum is to be plotted.
    epsilon : float
        The norm of the random matrices that are generated.
        Defaults to 10**-3
    num_pts : int
        The number of matrices, E, that will be used in the
        algorithm. Defaults to 20.
    '''
```

To get random matrices of the right norm, generate random matrices, divide them by their respective 2-norms, and then multiply by epsilon. Have your code plot the eigenvalues (point spectrum) of A in one color, and the point spectrum of each $A + E$ in a different color. The ϵ -pseudospectrum of A can be thought of as the region enclosed by your points. Test your code on

the 120x120 matrix:

$$\begin{pmatrix} 0 & i & -1 & 0 & \cdots & 0 \\ -i & 0 & i & -1 & \ddots & \vdots \\ 1 & -i & 0 & i & \ddots & 0 \\ 0 & 1 & -i & 0 & \ddots & -1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & i \\ 0 & \cdots & 0 & 1 & -i & 0 \end{pmatrix} \quad (1.3)$$

Your plot should resemble those in Figure 1.1.

This method of plotting, often called the “poorman’s pseudospectrum”, can only show the ϵ -pseudospectrum of A for one value of ϵ at a time, and even then it just provides a bound for the pseudospectrum. In many cases it does not give sufficient information, so we will now explain how to plot the ϵ -pseudospectrum using a more precise algorithm.

An Equivalent Pseudospectral Definition

Recall the following definition:

$$\sigma_\epsilon(A) = \{z \in \mathbb{C} : \|(zI_n - A)^{-1}\| > \epsilon^{-1}\}. \quad (1.4)$$

When we are using the induced 2-norm, we get that $\|(zI_n - A)^{-1}\|$ is the largest singular value of $(zI_n - A)^{-1}$. This is also defined for a matrix B as the square root of the largest eigenvalue of $B^H B$. This is equal to $1/s_{\min}(zI_n - A)$, where $s_{\min}(zI_n - A)$ is the smallest singular of $(zI_n - A)$. Hence, we can write

$$\sigma_\epsilon(A) = \{z \in \mathbb{C} \mid s_{\min}(A - zI_n) < \epsilon\}. \quad (1.5)$$

The Lanczos Method

Using this new definition for $\sigma_\epsilon(A)$, we can now plot pseudospectra using Lanczos iterations. Lanczos iterations make use of the Arnoldi method described in Lab ?? in order to more efficiently compute the smallest eigenvalue of $(zI_n - A)$. This method allows us to plot the pseudospectrum as a contour plot, where each contour corresponds to a fixed value of ϵ . The algorithm is provided below.

Note that while this algorithm gives better results than our previous method of plotting pseudospectra, it is much slower.

Problem 2. Finish the function below for implementing Algorithm 1.1 in Python. Make the function also plot the eigenvalues of the matrix on the same plot.

Algorithm 1.1 The Lanczos Method. This algorithm accepts a $n \times n$ square matrix, A ; a list of values for ϵ ; and an accuracy measure of m . It computes the $10^{-\epsilon}$ -pseudospectrum of A for each value of ϵ and produces the contour plot on the grid determined by $xvals$ and $yvals$ as returned by `psgrid()`.

```

1: procedure PSEUDOSPECTRUM( $A, m, \epsilon$ )
2:    $T \leftarrow \text{schur}(A)$  ▷ Compute the Schur decomposition of  $A$ 
3:    $eigsA \leftarrow \text{diagonal}(T)$ 
4:    $xvals, yvals \leftarrow \text{psgrid}(eigsA, m)$ 
5:    $sigmin \leftarrow \text{zeros}(m, m)$ 
6:   for  $k = 0 \dots m - 1$  do
7:     for  $j = 0 \dots m - 1$  do
8:        $T_1 \leftarrow (xvals[k] + i * yvals[j])I_n - T$ 
9:       ▷  $I_n$  is the  $n \times n$  identity matrix.
10:       $T_2 \leftarrow T_1^*$  ▷ Compute the conjugate transpose.
11:       $sigold \leftarrow 0$  ▷ Initialize variables.
12:       $qold \leftarrow \text{zeros}(n, 1)$ 
13:       $beta \leftarrow 0$ 
14:       $H \leftarrow \text{zeros}(n, n)$ 
15:       $q \leftarrow \text{random, complex-valued } n \times 1 \text{ array of norm } 1$ 
16:      ▷ Assume both real and imaginary parts are normally distributed
17:      for  $p = 0 \dots N - 2$  do
18:         $b_1 \leftarrow \text{solution to } T_2 x = q$ 
19:         $b_2 \leftarrow \text{solution to } T_1 x = b_1$ 
20:         $v \leftarrow b_2 - beta * qold$ 
21:         $alpha \leftarrow \text{real}(q^* * v)$ 
22:         $v \leftarrow v - alpha * q$ 
23:         $beta \leftarrow \|v\|$ 
24:         $qold \leftarrow q$ 
25:         $q \leftarrow v / beta$ 
26:         $H[p + 1, p] \leftarrow beta$ 
27:         $H[p, p + 1] \leftarrow beta$ 
28:         $H[p, p] \leftarrow alpha$ 
29:         $sig \leftarrow \text{absolute value of largest eigenvalue of } H[p + 1, : p + 1]$ 
30:        if  $|sigold / sig - 1| < 0.001$  then
31:          Break ▷ End loop if  $sig$  is close to  $sigold$ 
32:           $sigold \leftarrow sig$ 
33:           $sigmin[j, k] \leftarrow \sqrt{sig}$ 
34:      Plot the log of the values in  $sigmin$  as a contour plot on the grid determined
      by  $xvals$  and  $yvals$ .
```

```

def ps_contour_plot(A, m = 20, epsilon_vals=None):
    '''Plots the pseudospectrum of the matrix A as a contour plot. Also,
    plots the eigenvalues.

    Parameters:
        A : square, 2D ndarray
```

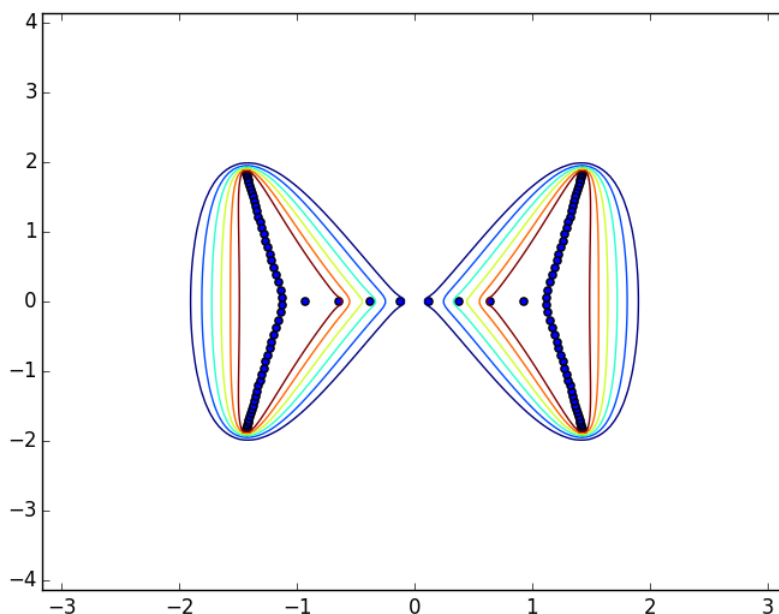


Figure 1.2: Contour plot of the ϵ -pseudospectrum of the matrix given in Problem 1, with $\epsilon = 10^{-2}, 10^{-3}, \dots, 10^{-7}$.

```

    The matrix whose pseudospectrum is to be plotted
    m : int
    accuracy
    epsilon_vals : list of floats
    If k is in epsilon_vals, then the epsilon-pseudospectrum
    is plotted for epsilon=10**(-k)
    If epsilon_vals=None, the defaults of plt.contour() are used
    instead of any specified values.
    ...

```

Test your code on the matrix from Problem 1. Your plot should look like Figure 1.2. You will need to choose a grid on which to plot the values computed in this algorithm. Use a grid that is roughly twice as large as one needed to plot the eigenvalues. To make your code run faster, make sure to use functions designed for Hermitian or triangular matrices whenever possible.

1. Plot the eigenvalues and contour lines on a complex plane.
2. m defaults to 20 but if you want the plot to look like Figure 1.2, you will need m to be close to 150. Note that this will take a while longer.

3. In order to plot the contour lines. Use the following code.

```
plt.contour(xvals,yvals,np.log10(sigmin), levels=epsilon_vals)
```

Hermitian, Normal, and Nonnormal Matrices

Recall that a matrix A is Hermitian if it is equal to its conjugate transpose, i.e. $A^H = A$. All the eigenvalues of Hermitian matrices are real. However, if the elements change slightly (adding E), these matrices quickly fall out of the class of Hermitian matrices, and the ϵ -pseudospectra may gain an imaginary component.

A broader class of matrices that includes all Hermitian matrices is the class of *normal matrices*. A normal matrix commutes with its conjugate transpose, i.e. $AA^H = A^HA$. In the case that A is normal, it can be shown that the ϵ -pseudospectrum consists of the union of circular sets about each of the eigenvalues of A . When A is not a normal matrix, the pseudospectrum can have more varied shapes. Because of this, plotting the pseudospectrum of a nonnormal matrix can provide more information about its behavior.

Figure 1.3 provides an example of this. The plots show the pseudospectra of the matrices:

$$B = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & i \end{pmatrix}, \quad B' = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & i \end{pmatrix}. \quad (1.6)$$

Note that these matrices have the same set of eigenvalues, but that while B is normal, B' is not. The plot of the pseudospectrum of B shows the expected circular regions, but the plot for B' lacks such symmetry. Another example of this can be seen in Figure 1.1. The pseudospectrum does not expand out from all of the eigenvalues evenly. Most of the region contained in the pseudospectrum lies on the sides of the plot, and not close to the center.

Because the pseudospectra of normal matrices behave so consistently, the study of pseudospectra focuses on nonnormal matrices. Pseudospectra can also be used to study infinite-dimensional linear operators, but different methods are required for plotting their pseudospectra.

Problem 3. Using your function from problem 1.3, investigate how the pseudospectra of Nonnormal, Hermitian, and Normal matrices differ. To do so, use the matrix given in problem 1.3 as the Nonnormal matrix.

For the Hermitian matrix, change the subdiagonal of 1s to be -1 .

For the Normal matrix, reset all of the subdiagonal entries to be zero (change the 1 diagonal and $-i$ diagonal to zeros). Set all of the diagonal entries to be random complex values (so as to get different eigenvalues).

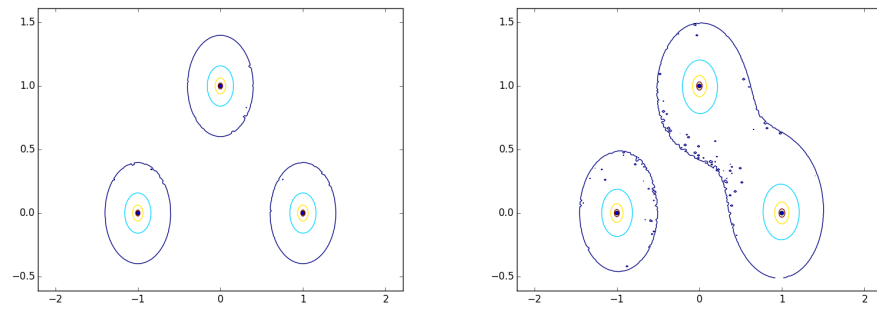


Figure 1.3: Plots of the Pseudospectrum of two matrices with the same eigenvalues. The plot on the left is for a normal matrix, and the one on the right is for a nonnormal matrix. The contours correspond to $\epsilon = 10^{-0.4}, 10^{-0.8}, 10^{-1.2}$

Plot your results.