Asymptotic Expansions for Bounds of ε-Pseudospectra of Nonnormal Matrices

by
ALEX BANK

Mihai Stoiciu, Advisor

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Mathematics

WILLIAMS COLLEGE
Williamstown, Massachusetts

April 30, 2020

Abstract

In this thesis, we first introduce the tenants of spectral theory and the concept of pseudospectra. We briefly discuss the relevant properties of pseudospectra and its connection to potential theory. Motivated by recent research on the pseudospectra of Jordan blocks, we study the pseudospectra of nonnormal matrices. Specifically, we look at the ε-pseudospectral radius of Jordan blocks and work to improve on the best known upper and lower bounds for these radii.

We define up with a new class of error matrix to produce the best known lower bound for the pseudospectral radius of the Jordan Block. We end with a computational approach to the problem by using software to approximate the pseudospectral radius and its expansion as a function of epsilon. The numbers generated in approximating the pseduospectral radius are new findings, and present a potential area and goal for further research.

Alex Bank
Williams College
Williamstown, Massachusetts 01267, USA

Acknowledgements

First, to my advisor Professor Mihai Stoiciu: my deepest thanks for your patience, kindness, and support. Besides this thesis, your Senior Seminar class and working as your TA have shown me just how deeply you care for your students and made my senior year beyond memorable. Thanks also to Professor Thomas Garrity for being my second reader—I appreciate all your insights and antics that have livened my math career. Thank you to all the professors in the department for making the math community at Williams such a wonderful place. I also want to thank the residents of Susan Hopkins house for the endless pots of tea that made this thesis possible. Lastly, I must thank my friends and family for all the times you kindly nodded and pretended to care while I went on about pseudospectra.

Contents

1 Introduction

Matrices exist everywhere in mathematics and everyday life. Any linear transformation—be it a rotation, reflection, or a translation—can be represented by a square matrix. However, most times just looking at the values of a matrix does not reveal the properties of the matrix.

Hence the concept of eigenvalues. Eigenvalues give personality to a matrix. The eigenvalues can be plotted and visualized, providing concrete information about the given matrix.

We will primarily explore the concept of matrix *pseudospectra* in this thesis. Nicholas Higam motivates the concept of pseudospectra very well in his review of a larger work by Lloyd Trefethen and Mark Embree [1]. Higam states that the study of eigenvalues began in the 1930s when several engineers were using matrix methods in an attempt to understand flutter (unwanted vibration) in aircrafts. Flutter is still an issue in aerodynamic design and testing. It turns out that the problem boils down to understanding the behavior of $\left\|e^{At}\right\|$ and $\left\|e^{A't}\right\|$ for a large matrix A. Pseudospectra explain the behavior of e^{At} and A^n as $t \to \infty$ and $t \to 0$, in particular by providing lower bounds that accurately track this behavior. More information can be found in the full Trefethen and Embree text [2].

The thesis is structured in three parts. The first part is mostly definitional and sets up key concepts for later explorations and proofs. The second section deals with known theory on the structure of pseudospectra of nonnormal matrices, and and the end of the section gets into original work on the 3×3 Jordan block. Finally, the last section contains new numerical work on the pseudospectral radii of Jordan blocks. The numerical analysis prompts several conjectures for further avenues of research.

2 Introduction to Pseudospectra

2.1 Spectra and Associated Definitions

Let A be an $N \times N$ matrix with real or complex coefficients; i.e. $A \in \mathbb{C}^{N\times N}$. Let $\vec{v}$ be a nonzero real or complex column vector of length N, and let λ be a real or complex scalar; we write $\vec{v} \in \mathbb{C}^N$ and $\lambda \in \mathbb{C}$. Then $\vec{v}$ is an *eigenvector* of A, and λ is its corresponding *eigenvalue*, if

$$A\vec{v} = \lambda\vec{v}. \tag{2.1}$$

The set of all eigenvalues of A is called the *spectrum* of A, denoted $\sigma(A)$, a nonempty subspace of the complex plane $\mathbb{C}$. We can equivalently define the spectrum as the set of points $z \in \mathbb{C}$ where the *resolvent* of the matrix,

$$(zI - A)^{-1} \tag{2.2}$$

does not exist. In this paper, we will use $z - A$ as shorthand for $zI - A$.

This second definition is useful to have because we will want to explore the behavior of $\|(z - A)^{-1}\|$ as a function of z. But to do this we first need to select a choice of norm, $\|\cdot\|$. Throughout this thesis we will use the 2-norm, defined by $\|\vec{x}\|_2 = (\sum |x_j|^2)^{1/2}$ for a vector $\vec{x} \in \mathbb{C}$ and by

$$\|A\|_2 = \max_{\vec{x}} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} \tag{2.3}$$

for a matrix A. We can alternatively think of equation 2.3 geometrically. If we apply our matrix A to every vector of magnitude one, the 2-norm equals the greatest magnitude of all the resulting vectors.

2.2 Pseudospectra

We start with the equivalent questions, "Is z an eigenvalue of A?" and "Is $z - A$ singular?" These are not robust questions, as an arbitrarily small perturbation of z can change the answer from yes to no. Therefore, we want to be asking the question, 'Is $\|(z - A)^{-1}\|$ large?' This question motivates the first definition of pseudospectra.

Definition 1: Pseudospectra

Let $A \in \mathbb{C}^{N\times N}$ and $\varepsilon > 0$ be arbitrary. The ε-pseudospectrum $\sigma_\varepsilon(A)$ of A is the set $z \in \mathbb{C}$ such that

$$\left\|(z-A)^{-1}\right\| > \varepsilon^{-1}$$

Another natural way to look at these ε-small perturbations is to consider the set of all matrices with norm less than epsilon. In other words, we can perturb our matrix by some other matrix with norm less than epsilon, and look at the spectrum of the resulting matrix. This motivates a second way to define pseudospectra.

Definition 2: Pseudospectra

The ε-pseudospectrum $\sigma_\varepsilon(A)$ is the set of $z \in \mathbb{C}$ such that

$$z \in \sigma(A+E)$$

for some $E \in \mathbb{C}^{N\times N}$ with $\|E\| < \varepsilon$.

One last concept we can consider is that of the *pseudo-eigenvector*. Notice that the equation $A\vec{v} = \lambda\vec{v}$ implies $(A-\lambda)\vec{v} = 0$, so the norm of the left hand side is zero. Thus, we may consider the pseudospectrum of A all of the points $z \in \mathbb{C}$ that get our norm very close to zero for a vector of length one, motivating the last definition for pseudospectra.

Definition 3: Pseudospectra

The ε-pseudospectrum $\sigma_\varepsilon(A)$ is the set of $z \in \mathbb{C}$ such that

$$\|(z-A)\vec{v}\| < \varepsilon$$

for some $\vec{v} \in \mathbb{C}^N$ with $\|\vec{v}\| = 1$.

Not surprisingly, we have the following theorem.

Theorem 2.1. *The three definitions above are equivalent.*

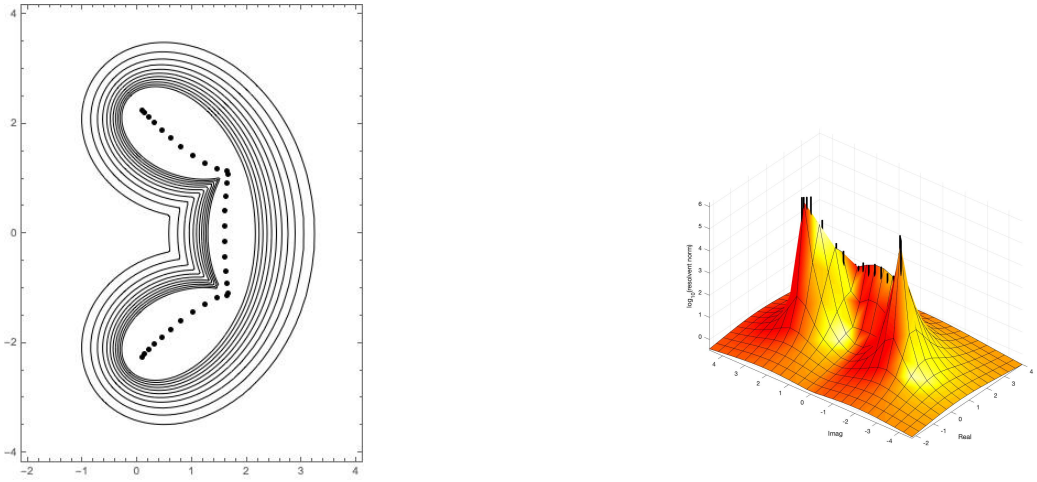
Proof. For $z \in \sigma(A)$, the equivalence is trivial, so assume $z \notin \sigma(A)$. This implies the existence of $(z - A)^{-1}$. To prove 2 $\Rightarrow$ 3, suppose that $(A + E)\vec{v} = z\vec{v}$ for some $E \in \mathbb{C}^{N\times N}$ with $\|E\| < \varepsilon$ and some nonzero $\vec{v} \in \mathbb{C}^N$, which we may say is normalized, or $\|\vec{v}\| = 1$. Then $\|(z - A)\vec{v}\| = \|E\vec{v}\| < \varepsilon$, as required. To prove 3 $\Rightarrow$ 1, suppose $(z - A)\vec{v} = s\vec{u}$ for some $\vec{v}, \vec{u} \in \mathbb{C}^N$ with $\|vecv\| = \|\vec{u}\| = 1$. and $s < \varepsilon$. Then $(z - A)^{-1}\vec{u} = s^{-1}\vec{v}$, so $\|(z - A)^{-1}\| \geq s^{-1} > \varepsilon^{-1}$. Finally, to prove 1 $\Rightarrow$ 2, suppose $\|(z - A)^{-1}\| > \varepsilon^{-1}$. Then $(z - A)^{-1}\vec{u} = s^{-1}\vec{v}$ and consequently $z\vec{v} - A\vec{v} = s\vec{u}$ for some $\vec{v}, \vec{u} \in \mathbb{C}^N$ each with norm equal to one and $s < \varepsilon$. It is enough to show that there exists a matrix $E \in \mathbb{C}^{N\times N}$ with $\|E\| = s$ and $E\vec{v} = s\vec{u}$, for then $\vec{v}$ will be an eigenvector of $A + E$ with eigenvalue z. In fact, E can be taken to be a rank-1 matrix of the form $E = s\vec{u}\vec{w}^*$ for some $\vec{w} \in \mathbb{C}^N$ with $\vec{w} * \vec{v} = 1$. In our case, $\|\cdot\|$ is the 2-norm, and this is evident simply by taking $\vec{w} = \vec{v}$. Thus, we have that the definitions are equivalent. □

Pseudospectra is nice in that it is inherently visual. We can plot the level curves defined by $\left\{z \in \mathbb{C}\middle|\|(A - z)^{-1}\| = \frac{1}{\varepsilon}\right\}$ for various values of $\varepsilon > 0$ and see how different sized perturbations change the spectrum of our matrix. The following example is the contour plot for the Grcar matrix defined by

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & & & & \\ -1 & 1 & 1 & 1 & 1 & & & \\ & -1 & 1 & 1 & 1 & 1 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -1 & 1 & 1 & 1 & 1 \\ & & & & -1 & 1 & 1 & 1 \\ & & & & & -1 & 1 & 1 \\ & & & & & & -1 & 1 \end{pmatrix}.$$

$$G = \begin{pmatrix} 2 & -1 & & & & & & -1 \\ & 2 & -1 & & & & & \\ & & 2 & -1 & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -1 & 1 & 1 & 1 & 1 \\ & & & & -1 & 1 & 1 & 1 \\ & & & & & -1 & 1 & 1 \\ & & & & & & -1 & 1 \end{pmatrix}.$$

This matrix is an a nonsymmetric Toeplitz matrix with notoriously-sensitive eigenvalues [3]. We will look at other Toeplitz matrices later on in this paper, but we bring up the Grcar matrix here because of the very beautiful pseudospectrum it produces.



Contour Plot 3D Plot

Figure 3: Psuedospectra of a Grcar matrix with $N = 32$

See the appendix of code for a robust Mathematica function to produce contour plots for any given matrix.

2.3 Properties of Pseudospectra

Before we begin our exploration of nonnormal matrices and their pseudospectra, it is helpful to list some of the properties of pseudospectra that explain why we are looking at nonnormal matrices.

First, we say a matrix $A \in \mathbb{C}^{N\times N}$ is *normal* if it has a complete set of orthogonal eigenvectors, that is, if it is unitarily diagonalizable,

$$A = UAU^*$$

where U is unitary. We have the following theorem regarding normal matrices.

Theorem 2.2. *Let $A \in \mathbb{C}^{N\times N}$ and $\|\cdot\| = \|\cdot\|_2$. A is normal if and only if*

$$\sigma_\varepsilon(A) = \sigma(A) + \Delta_\varepsilon,$$

where $\Delta_\varepsilon = \{z \in \mathbb{C} : |z| < \varepsilon\}$.

Proof. Suppose A is normal. Since we are using the 2-norm, it can be assumed without loss of generality to be diagonal without any effect on norms, with diagonal elements a_{jj} equal to the eigenvalues λ_j. In this case the resolvent is also diagonal, which implies that it satisfies the equation

$$\left\|(z-A)^{-1}\right\| = \frac{1}{\operatorname{dist}(z,\sigma A)}.$$

Notice $\sigma(A) + \Delta_\varepsilon = \{z : z = z_1 + z_2, z_1 \in \sigma(A), z_2 \in \Delta_\varepsilon\}$ which is equal to $\{z : \operatorname{dist}(z,\sigma(A)) < \varepsilon\}$. Thus, $\|(z-A)^{-1}\| > \varepsilon^{-1}$ for all $z \in \sigma(A) + \Delta_\varepsilon$, so $\sigma_\varepsilon(A) \supseteq \sigma(A) + \Delta_\varepsilon$. For the reverse inclusion, a proof can be found in [2, Theorem 2.2]. □

In the case of normal matrices, the pseudospectra is completely determined, and simply equal to the union of disks of radius epsilon centered at the eigenvalues. The following is a list of some other properties of pseudospectra.

Theorem 1: Properties of Pseudospectra

Let $A \in \mathbb{C}^{N\times N}$ and $\varepsilon > 0$ be arbitrary. Then

1. $\sigma_\varepsilon(A)$ is nonempty, open, and bounded, with at most N connected components, each containing one or more eigenvalues of A.
2. If $\|\cdot\| = \|\cdot\|_2$, then $\sigma_\varepsilon(A^*) = \overline{\sigma_\varepsilon(A)}$.
3. If $\|\cdot\| = \|\cdot\|_2$, then $\sigma_\varepsilon(A_1 \oplus A_2) = \sigma_\varepsilon(A_1) \cup \sigma_\varepsilon(A_2)$.
4. For any $c \in \mathbb{C}$, $\sigma_\varepsilon(A + c) = c + \sigma_\varepsilon(A)$.
5. For any nonzero $c \in \mathbb{C}$, $\sigma_{|c|\varepsilon}(cA) = c\sigma_\varepsilon(A)$.

A proof for items two through five can be found in [2, Theorem 2.4], however we will prove the first claim here after setting up some terminology. Let (X, d) be a metric space. Then $u : X \to [-\infty, \infty)$ is called *upper-semicontinuous* (at x) if $\limsup_{y\to x} u(y) \leq u(x)$ for all $x \in X$. We also let D be an open set in $\mathbb{C}$, then a function $u : D \to [-\infty, \infty)$ is called *subharmonic* (at w) if it is upper-semicontinuous and there exists $p > 0$ such that

$$u(w) \leq S_u^r(w) = \frac{1}{2\pi}\int_0^{2\pi} u(w + re^{it})dt \quad \text{for } 0 \leq r < p.$$

These definitions are from potential theory and are important because subharmonic functions follow the *maximum principle*.

Theorem 2.3 (Maximum Principle)**.** *Let u be a subharmonic function on a domain D in $\mathbb{C}$. Then if u attains a local maximum on D then u is constant.*

Full definitions and proofs of these concepts can be found in [4]. We will use this property in our proof by deriving a contradiction based on this principle.

Let us define the *resolvent map* for some matrix $A \in \mathbb{C}^{N\times N}$ as $R : \mathbb{C}\backslash\sigma(A) \to$

$\mathbb{C}^{N\times N}$ where $R(z) = (A - z)^{-1}$. Notice that we have the following equality,

$$\begin{aligned} R(z_1) - R(z_2) &= R(z_1)R(z_2)^{-1}R(z_2) - R(z_1)R(z_1)^{-1}R(z_2) \\ &= R(z_1)(A - z_2)R(z_2) - R(z_1)(A - z_2)R(z_2) \\ &= R(z_1)((A - z_2) - (A - z_1))R(z_2) \\ &= R(z_1)(z_1 - z_2)R(z_2). \end{aligned}$$

Rearranging this equation implies that

$$R(z) = [I - (z - z_0)R(z_0)]^{-1}R(z_0)$$

for a fixed $z_0 \in \mathbb{C} \setminus \sigma(A)$.

One tool we can use here is the Neumann series, which gives us the expansion $(1 - A)^{-1} = \sum_{n=0}^{\infty} A^n$ for any matrix A with norm less than one [5, 30]. This is a generalization of the geometric series in the real numbers. Notice that if we choose z to be very close to z_0, we have $\|(z - z_0)R(z_0)\| = |z - z_0|R(z_0)$, and thus can control the norm of this matrix to keep it less than one. Hence, we can rewrite 2.3 as

$$R(z) = \sum_{n=0}^{\infty}(z - z_0)^n R(z_0)^{n+1}$$

and thus, we conclude $R(z)$ is holomorphic in z with Taylor series $\left(\frac{d}{dz}\right)^n R(z) = n!R(z)^{n+1}$.

We use the Cauchy integral representation of $R(z)$. If C is taken to be the circle $|z - z_0| = r$, the formula may be written

$$R(z_0) = \frac{1}{2\pi i}\int_0^{2\pi} R(z_0 + re^{i\theta})d\theta.$$

Consequently we apply the norm to both sides of the equation and get

$$\begin{aligned} \|R(z_0)\| &= \left\|\frac{1}{2\pi i}\int_0^{2\pi} R(z + re^{i\theta})d\theta\right\| \\ &= \frac{1}{2\pi}\left\|\int_0^{2\pi} R(z + re^{i\theta})d\theta\right\| \end{aligned}$$

and finally using the triangle inequality, we obtain

$$\|R(z_0)\| \leq \frac{1}{2\pi} \int_0^{2\pi} \left\|R(z + re^{i\theta})\right\| d\theta.$$

which is exactly the form satisfying the definition of a subharmonic function. Now we are ready to prove the following theorem.

Theorem 2.4. *The ε-pseudospectrum of $A \in \mathbb{C}^{N\times N}$ is nonempty, open, and bounded, with at most N connected components each containing one or more eigenvalues of A.*

Proof. The ε-pseudospectrum of A contains the spectrum of A, which is necessarily nonempty. Thus, $\sigma_\varepsilon(A)$ is nonempty. The fact that the ε-pseudospectrum is open is a direct result of the First Definition of Pseudospectra, namely that the ε-pseudospectrum is the open subset of the complex plane bounded by the ε^{-1} level curve of the norm of the resolvent.

Another way to prove this is to use a theorem from real analysis that says for any function, $f : X \to Y$, that is continuous on X, the inverse function applied to an open set is open. We have the resolvent map as a continuous function from the complex plane with holes at the spectrum of A to the real numbers. The ε-pseudospectrum is just the inverse resolvent map of the open set $(1/\varepsilon, \infty)$. Thus, the ε-pseduospectrum is open in $\mathbb{C} \setminus \sigma(A)$.

Now suppose $\sigma_\varepsilon(A)$ has an open connected component that does not contain an eigenvalue of A. Let us call this component S. For all $s \in S$, we have $\|(A - s)^{-1}\| > 1/\varepsilon$. Since S is open and does not contain any eigenvalues of A, the maximum value of S must be contained on its interior. Let s_0 be the maximum point in S such that $\|(A - s_0)^{-1}\| \geq \|(A - s)^{-1}\|$ for all $s \in S$. However, this violates the maximum principle for subharmonic functions. Thus, our inequality implies that the norm of the resolvent is constant, and we have $\sigma_\varepsilon(A) = \mathbb{C}$. But since $\sigma(A)$ is nonempty and contained in the complex numbers, we have $\sigma(A) \subset S$, a contradiction. Thus, each connected component of $\sigma_\varepsilon(A)$ must contain and eigenvalue. Since there are at most N distinct eigenvalues, $\sigma_\varepsilon(A)$ has at most N connected components each containing one or more eigenvalues of A. □

This property of psuedospectra is quite apparent in 3D-plots. Looking at Figure 4 we can see that the pseudospectra forms tubes around the eigenvalues of

the matrix since the norm of the resolvent for an eigenvalue is defined as infinity. If we think about taking level curves in this plot, we can see that for smaller values of epsilon we will get three connected components centered at the eigenvalues, and as we let epsilon increase, the components merge together.

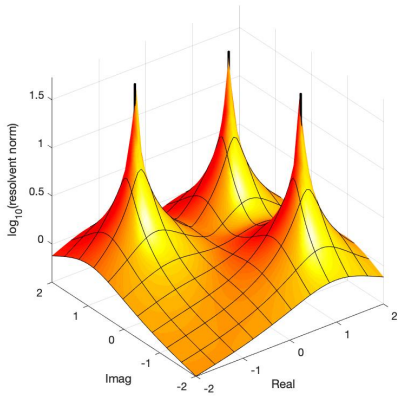


Figure 4: 3D plot for a 3×3 matrix

2.4 General Bounds for Diagonalizable Matrices

We begin here the characterization of the behavior of nonnormal, diagonalizable matrices. Write $A = VDV^{-1}$. We define the *condition number* of V as

$$\kappa(V) = \|V\| \left\|V^{-1}\right\| = \frac{s_{\max}(V)}{s_{\min}(V)}$$

where $s_{\max}(V)$ and $s_{\min}(V)$ are the maximum and minimum singular values of V, respectively. One caution is that there is ambiguity in the condition number, as V is not uniquely determined. The condition number becomes unique if the eigenvalues are distinct and the columns of V are normalized by $\|v_j\| = 1$. Van Der Sluis showed that this choice may not necessarily be the one that minimized $\kappa(V)$, but this choice exceeds the minimal value by at most a factor of $\sqrt{N}$ [6].

Notice $\kappa(V)$ is necessarily greater than or equal to 1. Moreover, $\kappa(V) = 1$ if and only if A is normal [7]. The importance of the condition number is evident in the following theorem that gives an upper and lower bound on the size of the pseudospectrum of a nonnormal, diagonalizable matrix.

Theorem 2.5 (Bauer-Fike)**.** *Let $A \in \mathbb{C}^{N\times N}$ and let A be diagonalizable, such that $A = VDV^{-1}$. Then for each $\varepsilon > 0$,*

$$\sigma(A) + B(0,\varepsilon) \subseteq \sigma_\varepsilon(A) \subseteq \sigma(A) + B(0,\varepsilon\kappa(V))$$

where

$$\kappa(V) = \|V\| \left\|V^{-1}\right\| = \frac{s_{max}(V)}{s_{min}(V)}.$$

The original proof for this can be found in [8]. We can see that the lower bound is a union of balls of radius epsilon around the eigenvalues of A, or exactly the pseudospectrum if A was normal. Clearly nonnormal matrices are much more sensitive to perturbations, and in later sections we will relax the restriction that our matrix is diagonalizable and work to find other upper and lower bounds for these other classes of matrices.

3 Pseudospectra of Nonnormal Matrices

Now that we have an understanding of properties of pseudospectra and the knowledge that normal matrices have completely determined pseudospectrum, we can turn towards understanding nonnormal matrices. Most of the work being done is finding upper and lower bounds for the pseudospectra; if we can find better bounds for pseudospectra, then we can approximate things like the radius of pseudospectra and identify special cases. Serval types of nonnormal matrices are well understood, but there is no unifying theory. In this section, we will look at serval types of nonnormal matrices and their resulting pseudospectrum.

3.1 Circulant Matrices

We begin our exploration with *circulant matrices*. A matrix $C \in \mathbb{C}^{N\times N}$ is circulant if it has the form

$$C = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix}.$$

Circulant matrices are nice. One amazing property of circulant matrices is that the eigenvectors are always the same. The eigenvalues are different for each C, but since we know the eigenvectors they are easy to diagonalize. We can actually see one eigenvector right away:

$$\vec{v}_0 = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

This is an eigenvector because multiplying $C\vec{v}_0$ simply sums each row of C. But since each row of C contains the same entries, we have

$$C\vec{v}_0 = \underbrace{(c_0 + c_1 + \cdots + c_{n-1})}_{\lambda_0}\vec{v}_0$$

and thus, λ_0 is an eigenvalue of C. So we have one eigenvalue, but what about the others? The others turn out to be very easy to write down thanks to a very special value: a *primitive roots of unity*:

$$\omega_n = e^{\frac{2\pi i}{n}}$$

The quantity ω_n has the property that $\omega_n^n = e^{2\pi i} = 1 = \omega_n^0$, but no smaller power equals one. Therefore, $\omega_n^{j+n} = \omega_n^j$, so the exponents are periodic. Using this quantity, we can write the equation for the k-th eigenvector of C ($k = 0, 1, \ldots, n-1$):

$$\vec{v}_k = \begin{pmatrix} \omega_n^{0k} \\ \omega_n^{1k} \\ \omega_n^{2k} \\ \vdots \\ \omega_n^{(n-1)k} \end{pmatrix}$$

and we have the corresponding eigenvalues:

$$\lambda_k = c_0 + c_{n-1}\omega_n^k + c_{n-2}\omega^{2k} + \cdots + c_1\omega^{n-1}.$$

Let's see why this works. Let $\vec{y} = C\vec{v}_k$. The the ℓ-th component of $\vec{y}$ is

$$\vec{y}_\ell = \sum_{j=0}^{n-1} c_{j-\ell}\omega_n^{jk} = \omega_n^{\ell k} \sum_{j=0}^{n-1} c_{j-\ell}\omega_n^{(j-\ell)k}$$

Notice the remaining sum is now independent of ℓ because both c_j and ω_n^j are periodic. In other words, moving from j to $j - \ell$ only re-arranges the numbers being summed (a circular shift, hence the name of the matrix), so you get the same sum for each component of $\vec{y}$. Thus,

$$C\vec{v}_k = \lambda_k\vec{v}_k, \quad \text{where } \lambda_k = \sum_{j=0}^{n-1} c_j\omega_n^{jk}.$$

Because of the periodic nature of the roots of unity, the eigenvectors are independent and form an orthonormal basis, implying circulant matrices are normal. Because of this fact, the pseudospectra of circulant matrices are completely

described. However, circulant matrices are important to our later discussions. First, circulant matrices are a special type of Toeplitz matrix, and we will see more Toeplitz matrices in the next section.

The other reason will come up later when we look at error matrices. When choosing error matrices to look at, we want to either be able to control the norm of our chosen error matrix, or control the eigenvalues of our perturbed matrix. Thus, we will look at circulant matrices since they give us the ability to control eigenvalues.

3.2 Triangular Toeplitz Matrices

We begin by defining our matrix A as an upper triangular Toepliz matrix defined by coefficients $a_k \in \mathbb{C}$, $0 \leq k \leq N-1 \leq \infty$,

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{N-1} \\ & a_0 & a_1 & \ddots & \vdots \\ & & a_0 & \ddots & a_2 \\ & & & \ddots & a_1 \\ & & & & a_0 \end{pmatrix}$$

and let $f(z)$ be the *symbol* of this matrix, defined by

$$f(z) = \sum_{k=0}^{N-1} a_k z^k.$$

Because the matrix is triangular, the spectrum of A is well understood and simply equal to the terms on the diagonal. In other words,

$$\sigma(A) = f(\{0\}) = \{a_0\}.$$

So we have the characterization of the spectra, but what can we say about the pseudospectra? Work done by Reichel and Trefethen has established an upper and lower bound for these triangular Toeplitz matrices [9]. While the theorem is stated for upper triangular Toeplitz matrices, the same results hold for the lower triangular case because the identity $\sigma_\varepsilon(A) = \sigma_\varepsilon(A^T)$ that holds for any matrix A and any ε.

Theorem 3.1. *Let A_N be an $N \times N$ (nondiagonal) triangular Toeplitz matrix with entries $\{a_k\}$ and symbol $f_N(z)$. Let Δ be the closed unit disk and Δ_r be the closed disk of arbitrary radius r. If c_N and r are defined by*

$$c_N = \sum_{k=1}^{N-1} |a_k| > 0, \qquad r = \left(\tfrac{\varepsilon}{c_N}\right)^{1/N},$$

then for any $\varepsilon \geq 0$,

$$f_N(\Delta_r) \subseteq \sigma_\varepsilon(A_N) \subseteq f_N(\Delta) + \Delta_\varepsilon.$$

In most cases we are taking $\varepsilon \ll c_N$, and thus $r < 1$. This means that establishing the lower bound the more important part of this theorem since the upper bound is a fixed value plus a small epsilon radius, while the lower bound essentially depends of epsilon. We will prove the first inclusion of this theorem.

Proof. To prove the first inclusion, our goal is to construct a pseudo-eigenvector and use Definition 3 to show containment in the epsilon-pseudospectrum. Given $\varepsilon > 0$, let r be defined as in 3.1. Now, given any $\lambda \in f_N(\Delta_r)$, let $\lambda = f_N(z)$ for some $z \in \Delta_r$ and define $u = (1, z, z^2, \ldots, z^{N-1})^T$. Then we have the following equality

$$(\lambda I - A_N)u = z^N \begin{pmatrix} 0 & & & & & \\ a_{N-1} & \cdot & & & & \\ \vdots & \cdot & \cdot & & & \\ a_3 & & \cdot & \cdot & & \\ a_2 & a_3 & & \cdot & \cdot & \\ a_1 & a_2 & a_3 & \cdots & a_{N-1} & 0 \end{pmatrix} u.$$

This becomes clear when you substitute $\lambda = f_N(z) = \sum_{k=o}^{N-1} a_k z^k$. Then the algebra works out nicely. The matrix on the right hand side has its norm bounded by c_N. This implies

$$\frac{\|(\lambda I - A_N)u\|}{\|u\|} \leq |z|^N c_N \leq r^N c_N = \frac{\varepsilon}{c_N} c_N = \varepsilon$$

so u is an ε-pseudoeigenvector of A_N and thus $\lambda \in \sigma_\varepsilon(A_N)$ by Definition 3. Hence we have the inclusion $f_N(\Delta_r) \subseteq \sigma_\varepsilon(A_N)$. □

3.3 The Jordan Block

Now we want to consider the simplest possible case of triangular Toeplitz matrices (and probably the simplest possible nonnormal matrix), the Jordan block, defined

by the $N \times N$ matrix

$$J_N = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ & & & & 0 \end{pmatrix}.$$

The Jordan block has the corresponding symbol $f(z) = z$, which we will use when applying theorem 3.1 to find a lower bound for the pseudospectra. However, we first must note one curious thing about the pseudospectra of the Jordan block; despite being a nonnormal matrix, the pseudospectrum appears to be a disk.

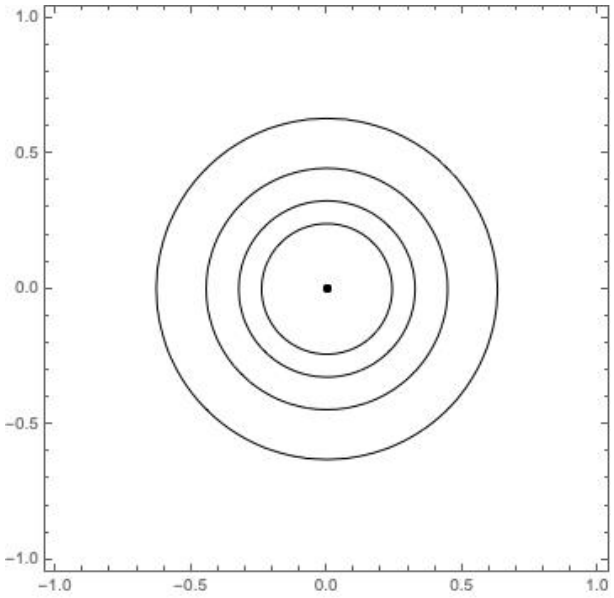


Figure 5: Level curves for the pseudospectrum of the 4×4 Jordan block

Indeed, it is the case that the pseudospectrum of the Jordan block is a disk.

Theorem 3.2. *The ε-pseudospectrum of a Jordan block J_N is a disk.*

To prove this, we need the following lemma.

Lemma 3.3. *Let A, B be $n \times n$ matricies with complex coefficients. If there exists a unitary matrix, U, such that $A = UBU^*$, then $\|A\| = \|B\|$.*

The proof of this lemma relies on the property of unitary matrices that they

preserve norms; in other words, $\|U\| = 1$. Hence we have the following equations:

$$\|A\| = \|UBU^*\| \leq \|U\| \, \|B\| \, \|U^*\| = \|B\|$$
$$\|B\| = \|U^*AU\| \leq \|U^*\| \, \|A\| \, \|U\| = \|A\|$$

Thus, $\|A\| = \|B\|$. Now we proceed with the proof of theorem 3.2.

Proof. To prove $\sigma_\varepsilon(J_N)$ is a disk, we will first prove that it is rotation invariant, then prove any line drawn between the eigenvalue and some $\lambda \in \sigma_\varepsilon(J_N)$ is in the pseudospectrum. We will first show that if $\lambda \in \sigma_\varepsilon(J_N)$, then $e^{i\theta}\lambda \in \sigma_\varepsilon(J_N)$. First, consider the matrix

$$S = \begin{pmatrix} e^{i\theta} & & & \\ & e^{2i\theta} & & \\ & & \ddots & \\ & & & e^{ni\theta} \end{pmatrix}$$

for any $\theta \in [0, 2\pi)$. Note that this matrix is unitary since

$$SS^* = \operatorname{diag}(|e^{i\theta}|^2, |e^{2i\theta}|^2, \ldots, |e^{ni\theta}|^2) = I.$$

More importantly, we have that $SJ_NS^* = e^{i\theta}J_N$. In other words, S rotates J_N by theta.

Suppose we have some $\lambda \in \sigma_\varepsilon(J_N)$. Then there exists some E with $\|E\| < \varepsilon$ such that $\lambda \in \sigma(J + E)$ by Definition 2 of pseudospectra. Thus we have some corresponding eigenvector, $\vec{v}$, such that

$$(J + E)\vec{v} = \lambda\vec{v}.$$

Multiplying by e^{ix}, we obtain

$$(e^{ix}J + e^{ix}E)\vec{v} = e^{ix}\lambda\vec{v}$$
$$(\tilde{J} + e^{ix}E)\vec{v} = (e^{ix}\lambda)\vec{v}.$$

Notice that $\|e^{ix}E\| = |e^{ix}| \, \|E\| = \|E\| < \varepsilon$. Thus, $e^{ix}\lambda \in \sigma_\varepsilon(\tilde{J})$, and since $\sigma_\varepsilon(J) = \sigma_\varepsilon(\tilde{J})$, we have $e^{ix}\lambda \in \sigma_\varepsilon(J)$ as desired.

Now that we have rotation invariance, we must prove that every point inside the disk is also in the pseudospectrum. Suppose this is not the case; in other words,

suppose for some $\lambda \in \sigma_\varepsilon(J_N)$ there exists some $t \in (0,1)$ such that $t\lambda \notin \sigma_\varepsilon(J_N)$. The we have the sets $U = D(0, |t\lambda|)$ and $V = \mathbb{R}^2 \setminus \overline{D(0, |t\lambda|}$. We have that $U \cap \sigma_\varepsilon(J) \neq \emptyset$ since the eigenvalue zero lies in this set. Similarly, $V \cap \sigma_\varepsilon(J) \neq \emptyset$ since λ lies in this set. Hence we have $\sigma_\varepsilon(J) = (U \cap \sigma_\varepsilon(J)) \cup (V \cap \sigma_\varepsilon(J))$, which implies that the ε-pseudospectrum of J is not connected, a contradiction of 2.4.

Thus, the pseudospectrum of a Jordan block is a disk. □

When we apply Theorem 3.1 to our Jordan block, we have $c_N = 1$ and $r = \varepsilon^{1/N}$, thus getting a lower bound on the pseudospectral radius of $r_N(\varepsilon) = \varepsilon^{1/N}$. The question now is can we do better? One way at getting better bounds is to use classes of error matrices, E, where we can both control the norm and come up with characteristic equations for $J_N + E$ that are easy to solve. Consider the matrix

$$E = \begin{pmatrix} 0 & \varepsilon & & & \\ & 0 & \varepsilon & & \\ & & 0 & \ddots & \\ & & & \ddots & \varepsilon \\ \varepsilon & & & & 0 \end{pmatrix}$$

then $\|E\| = \varepsilon$. Then for any $\lambda \in \sigma(J_N + E)$, we have

$$0 = \det(\lambda I - (J_N + E)) = (-1)^N \lambda^N + (-1)^{N+1} \varepsilon (1+\varepsilon)^{N-1}$$

After some algebra, we solve for lambda and get

$$\lambda = \sqrt[N]{\varepsilon(1+\varepsilon)^{N-1}} \tag{3.1}$$

so we have a better lower bound for the pseudospectral radius, and have proved the following theorem [7].

Theorem 3.4 (Better Lower Bound)**.** *Let J be an $N \times N$ Jordan block. Then*

$$B(0, \sqrt[N]{\varepsilon(1+\varepsilon)^{N-1}}) \subseteq \sigma_\varepsilon(J_N).$$

In a later section, we will explore the radius numerically using software and see how well this lower bound fares against the numbers we find.

3.4 The 2×2 Jordan Block

We will start by looking at the smallest Jordan block, the 2×2 case. What is notable about this case is that we have a closed form for the norm of the matrix.

Let $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, with $a, b, c, d \in \mathbb{C}$. Let $s_{\max}$ denote the largest singular value of A.

Then,

$$\|A\|^2 = s_{\max} \tag{3.2}$$

$$= \text{largest eigenvalue of } A^*A \tag{3.3}$$

$$= \frac{\operatorname{Tr}(A^*A) + \sqrt{\operatorname{Tr}(A^*A)^2 - 4\det(A^*A)}}{2} \tag{3.4}$$

Letting $\rho = |a|^2 + |b|^2 + |c|^2 + |d|^2$, we can expand the norm as

$$\|A\|^2 = \frac{1}{2}\left(\sqrt{\rho^2 - 4\big(|a|^2|d|^2 + |b|^2|c|^2 - (ad\overline{bc} + \overline{ad}bc)\big)} + \rho\right).$$

This formula allows us to explicitly compute the ε-pseudospectrum of the 2×2 matrices and the following proposition.

Proposition 3.5. *Let A be any non-diagonalizable 2×2 matrix, and let λ denote the eigenvalue of A. Write $A = VJV^{-1}$ where*

$$V = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad a, b, c, d \in \mathbb{C}, \quad \textit{and} \quad J_2 = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}.$$

Then $\sigma_\varepsilon(A)$ is exactly a disk. More specifically, given any ε,

$$\sigma_\varepsilon(A) = B(\lambda, |k|)$$

where

$$|k| = \sqrt{C\varepsilon + \varepsilon^2} \quad \textit{and} \quad C = \frac{|a|^2 + |c|^2}{|ad - bc|}. \tag{3.5}$$

A full proof of this proposition can be found in [7], but is essentially a direct calculation of $\|(z - A)^{-1}\|$ using equation 3.4. When we apply this proposition to our 2×2 Jordan block we have V is the identity matrix so $a = d = 1$ and $b = c = 0$,

and thus plugging into 3.5 we get that

$$|k| = \frac{\sqrt{\varepsilon^2(1-0)(1-0)+\varepsilon|1-0|(1+0)}}{|1-0|} = \sqrt{\varepsilon+\varepsilon^2}.$$

Hence,

$$\sigma_\varepsilon(J_2) = B(\lambda, \sqrt{\varepsilon+\varepsilon^2}).$$

Notice that this is exactly the same radius proposed by Theorem 3.4 when $N = 2$. However, this is a special case, and we will see in the next section that the Better Lower Bound is only the exact radius in the 2×2 case.

3.5 The 3×3 Jordan Block

In this section we consider the pseudospectra of the 3×3 Jordan block. Unlike the 2×2 Jordan block, there is no closed formula for the norm, so our investigation will focus on finding lower bounds on the ε-pseudospectral radius. Using the formula established in Theorem 3.4 we have the disk of radius

$$r = \sqrt[3]{\varepsilon(1+\varepsilon)^{(3)-1}} = \sqrt[3]{\varepsilon+2\varepsilon^2+\varepsilon^3}$$

contained in the ε-pseudospectrum, our current best lower bound. Here, we find another type of error matrix to produce a lower bound on the ε-pseudospectra.

Lemma 3.6. *Let*

$$E = \begin{pmatrix} j & \nu & 0 \\ 0 & k & \nu \\ \nu & 0 & \ell \end{pmatrix}$$

where $j = \mu^{1/3}$, $k = -e^{i\pi/3}\mu^{1/3}$, *and* $\ell = e^{2i\pi/3}\mu^{1/3}$. *We also require* $\mu > 0$ *and* $\nu > 0$.
Then $\|E\| = \mu^{1/3} + \nu$ *and for* $\|E\| < \varepsilon$, *we have*

$$B(0, \sqrt[3]{\mu+\nu(\nu+1)^2}) \subseteq \sigma_\varepsilon(A)$$

Proof. Let

$$E = \begin{pmatrix} j & \nu & 0 \\ 0 & k & \nu \\ \nu & 0 & l \end{pmatrix}$$

where $j = \mu^{1/3}$, $k = -e^{i\pi/3}\mu^{1/3}$, and $l = e^{2i\pi/3}\mu^{1/3}$. We also require $\mu > 0$ and $\nu > 0$. Then we have the following equation for the norm of E:

$$\begin{aligned}\|E\|^2 &= \text{largest eigenvalue of } E^*E \\ &= \text{largest eigenvalue of } \begin{pmatrix} \mu^{2/3}+\nu^2 & \mu^{1/3}\nu & e^{2i\pi/3}\mu^{1/3}\nu \\ \mu^{1/3}\nu & \mu^{2/3}+\nu^2 & e^{2i\pi/3}\mu^{1/3}\nu \\ -e^{i\pi/3}\mu^{1/3}\nu & -e^{i\pi/3}\mu^{1/3}\nu & \mu^{2/3}+\nu^2 \end{pmatrix}.\end{aligned}$$

We get the characteristic polynomial

$$0 = \det(E^*E - \lambda I) = (\mu^{2/3} - \mu^{1/3}\nu + \nu^2 - \lambda)^2(\mu^{2/3} + 2\mu^{1/3}\nu + \nu^2 - \lambda)$$

Thus we have the following two solutions for lambda,

$$\begin{aligned}\lambda &= \mu^{2/3} - \mu^{1/3}\nu + \nu^2 \\ \lambda &= \mu^{2/3} + 2\mu^{1/3}\nu + \nu^2 = (\mu^{1/3} + \nu)^2\end{aligned}$$

We now have $\|E\| = \max(\sqrt{\mu^{2/3} - \mu^{1/3}\nu + \nu^2}, \mu^{1/3} + \nu)$, and since $\mu > 0$ and $\nu > 0$ we have $\|E\| = \mu^{1/3} + \nu$.

Choosing μ and ν such that $\|E\| < \varepsilon$, we can use the second definition of pseudospectra to say $\sigma(J_3 + E) \subseteq \sigma_\varepsilon(J_3)$. Looking at the characteristic polynomial of $J_3 + E$, we have

$$\begin{aligned}0 &= \det(J_3 + E - \lambda I) \\ &= \det \begin{pmatrix} j-\lambda & \nu+1 & 0 \\ 0 & k-\lambda & \nu+1 \\ \nu & 0 & l-\lambda \end{pmatrix} \\ &= (j-\lambda)(k-\lambda)(l-\lambda) + \nu(\nu+1)(\nu+1) \\ &= \mu - \lambda^3 + \nu(\nu+1)^2 \\ \lambda &= \sqrt[3]{\mu + \nu(\nu+1)^2}\end{aligned}$$

Thus, we get that $B(0, \sqrt[3]{\mu + \nu(\nu+1)^2}) \subseteq \sigma_\varepsilon(J_3)$. This method gives us a classes of lower bounds. □

For example, let $0 < k < \varepsilon$ and $\alpha \geq 3$. Let $\mu = k^\alpha$ and $\nu = k - k^{\alpha/3}$. We will call this a perturbation of type A. For small values of epsilon, we have that $\|E\| = k^{\alpha/3} + (k - k^{\alpha/3}) = k < \varepsilon$. This type A perturbation gives us the following equation for the pseudospectral radius:

$$r_3(\varepsilon) \geq \sqrt[3]{\varepsilon^3 - \varepsilon^{\alpha/3} + 2\varepsilon^{2\alpha/3} + 2\varepsilon^2 - 3\varepsilon^{(6+\alpha)/3} + \varepsilon - 4\varepsilon^{(3+\alpha)/3} + 3\varepsilon^{(3+2\alpha)/3}}.$$

The interesting thing about this expansion is the different equations yielded by changing our tuning parameter, α. The following table gives us a few examples of the expansions we obtain with different values of α

α	Lower Bound of the Radius Cubed
3	ε^3
4	$\varepsilon - \varepsilon^{4/3} + 2\varepsilon^2 - 4\varepsilon^{7/3} + 2\varepsilon^{8/3} + \varepsilon^3 - 3\varepsilon^{10/3} + 3\varepsilon^{11/3}$
5	$\varepsilon - \varepsilon^{5/3} + 2\varepsilon^{8/3} + \varepsilon^3 + 2\varepsilon^{10/3} - 3\varepsilon^{11/3} + 3\varepsilon^{13/3}$
6	$\varepsilon + \varepsilon^2 - 3\varepsilon^3 - \varepsilon^4 + 3\varepsilon^5$

The accompanying plot compares these expansions with the Better Lower Bound for a 3×3 Jordan block 6.

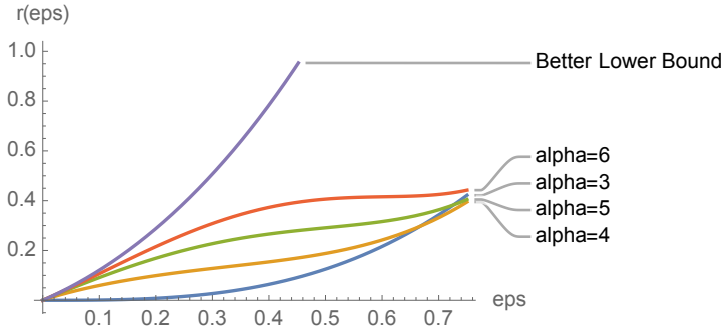


Figure 6: A comparison of Lower Bounds

In every case the Better Lower Bound is still greater than our new expansions. As $\alpha \to \infty$, we have $\mu \to 0$ and $\nu \to k$ so we end up with the same radius as what we found in the Better Lower Bound. This trend is even more explicit when we zoom in closer to zero. In Figure 7 we can see how increasing alpha brings us closer and closer to our original lower bound. What is interesting about these expansions is that we get epsilon raised to fractional powers when α is not a multiple of three. We will see in the next section that fractional exponents may be required

to capture the exact nature of the pseudospectral radius. Thus, error matrices with two parameters such as this type A error that can balance exponents should be explored further to continue to improve upon the best known lower bound.

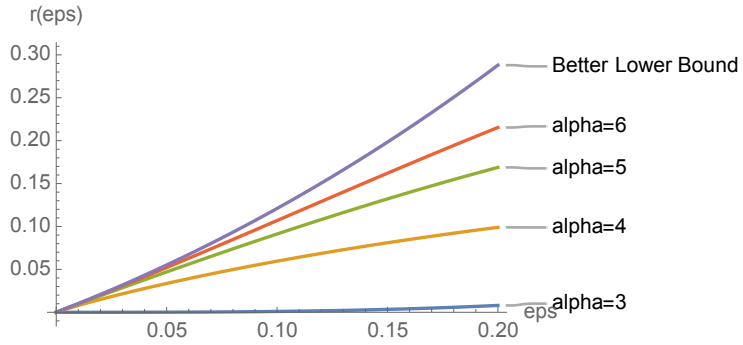


Figure 7: Comparison at epsilon close to zero

4 Computational Analysis of Pseudospectral Radius

Now that we understand classes of lower bounds on the ε-pseudospectra for different sized Jordan blocks, we will use computer software to explore pseudospectra and calculate the radius for the different sized Jordan blocks. Then we can compare our numerical results with the lower bounds we found earlier and see how well they approximate the radius.

4.1 Radius of the Jordan Block

We know the case $N = 2$ has an exact solution for the radius thanks to the closed-form equation for $\|(z - J_2)^{-1}\|$. Namely, we obtain the solution $\sigma_\varepsilon(J_2) = B(0, \sqrt{\varepsilon - \varepsilon^2})$. It is important to keep this equation in mind because we can use it as a litmus test for any numeric analysis we run; we must produce this radius with our analysis or else it is a bad test. For higher values of N, there is no closed-form equation for the norm. However, from [10] we have the following:

$$\lim_{\varepsilon \to 0} r_N(\varepsilon) = \sqrt[N]{\varepsilon}. \tag{4.1}$$

Equivalently, we can take the Nth power of each side and say $r_N(\varepsilon)^N = \varepsilon$ as $\varepsilon \to 0$. The $N = 2$ case and theorem 3.4 show us there may be higher order terms in epsilon that produce better approximations of the radius. We want to explore the *Puiseux series* for $r_N(\varepsilon)$. This definition follows from [11, p. 130]

Definition: Puiseux Series

The Puiseux series is a generalization of power series that allow for negative and fractional exponents. If K is a field then we say the field of Puiseux series with coefficients in K is the set of expressions of the form

$$f = \sum_{k=k_0}^{\infty} c_k x^{k/N}$$

where N is a positive integer and k_0 is an arbitrary integer.

A formal Puiseux series is a power series with rational exponents which share some natural number N as their common denominator. We have the componentwise addition, and the usual convolution product, which is defined by

$$\Big(\sum_{k=k_0}^{\infty} c_k x^{k/N}\Big)\Big(\sum_{k=k_0}^{\infty} d_k x^{k/N}\Big) = \sum_{r=0}^{\infty}\Big(\sum_{s=0}^{r} c_s d_{r-s}\Big)x^{k/N}.$$

The classic motivation for Puiseux series comes from algebraic geometry with application in tropical geometry. When working over an algebraically closed field such as the complex numbers, Puiseux series are also algebraically closed [12].

We know the first term in the Puiseux series for $r_N(\varepsilon)^N$ is ε. Using an arbitrary expression for the next unknown term we can write

$$r_N(\varepsilon)^N = \varepsilon + c\varepsilon^p$$

for some $c, p \in \mathbb{R}$ since we know the pseudospectra is a disk, and thus will have a real-valued radius. By subtracting away the known terms and rearranging, we get the following equation:

$$\log(r_N(\varepsilon)^N - \varepsilon) = \log(c\varepsilon^p) \tag{4.2}$$

$$= \log(c) + p\log(\varepsilon) \tag{4.3}$$

By taking various values of epsilon, we can fit a linear polynomial using the log of epsilon and the log of the terms we have already solved for and the slope of the line will give us the value for p and the exponential of the intercept will give us the value for c. After having solved these terms, we fill in the values for c and p, and repeat the process on the next unknown term.

Let `eps` be a vector of epsilon values, `radEps` be the vector of corresponding radii, `n` be the dimension of the Jordan block, and `knownTerms` be an $N \times 2$ matrix with known constants in the first column and known powers in the second column. Then we produce the following MATLAB algorithm:

Algorithm 1: `findNextPower`

Input: eps, radEps, n, knownTerms

Result: (c, p)

```
x = log(eps);
d = size(knownTerms);
knownSum = zeros(1, length(eps));
for i = 1:d(1)
     tempEps = arrayfun(@(x) x^(knownTerms(i,2)), eps);
     knownSum = knownSum + knownTerms(i,1)*tempEps;
end
radPower = arrayfun(@(x) x^n, radEps);
y = log(radPower - knownSum);
fit = polyfit(x,y,1);
p = fit(1);
c = exp(fit(2));
```

We will walk through this algorithm step-by-step for some $A \in \mathbb{C}^{N\times N}$. First suppose we only have one value for epsilon, ε_1, and the radius for the corresponding epsilon-pseudospectrum, $r_N(\varepsilon_1)$. Suppose we also have the first d terms of the Puiseux series for the radius,

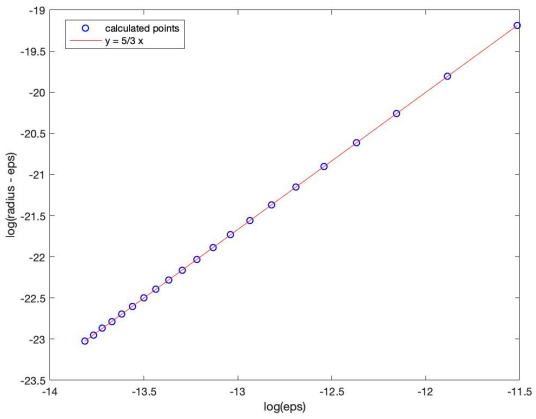
$$\tilde{r}_N(\varepsilon)^N = \sum_{i=0}^{d} c_{k_i}\varepsilon^{k_i/N}.$$

The algorithm calculates two values: $\log(\varepsilon_1)$ and $\log(r_N(\varepsilon_1)^N - \tilde{r}_N(\varepsilon_1)^N)$. From our equation above, 4.3, we can see that in the linear trend for the log of the Pusieux series $\log(\varepsilon_1)$ acts as our independent variable and $\log(r_N(\varepsilon_1)^N - \tilde{r}_N(\varepsilon_1)^N)$ is our dependent variable. Now suppose we have many epsilon values, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$, and their corresponding radii, $r_N(\varepsilon_1), r_N(\varepsilon_2), \ldots, r_N(\varepsilon_m)$. The algorithm calculates and stores $\log(\varepsilon_i)$ and $\log(r_N(\varepsilon_i)^N - \tilde{r}_N(\varepsilon_i)^N)$ for each $i \in \{1, 2, \ldots, m\}$. Now we fit a line using these points; the `polyfit` function in MATLAB finds a coefficient and intercept for a best-fit line that minimizes the squared distance of each point to the line. Once we raise each side of our linear equation to an exponential, we end up with the exponential of our linear intercept as the next coefficient in the series,

and our linear coefficient becomes our next exponent in the Puiseux series. These are the terms returned by the function.

The `Eigtool` package in MATLAB provides functions for computing the ε-pseudospectral radius. One issue with the algorithm is that the returned coefficients are the result of fitting a line to a sequence of points that are only linear as epsilon approaches zero. The nice thing is that the algorithm self-corrects; if some coefficient, say c_i corresponding to the exponent p_i, is incorrect, then the algorithm will return a new coefficient and the same exponent p_i the next time we run the algorithm.

Consider an example of how the algorithm functions finding the second term in the radius for $\sigma_\varepsilon(J_3)$. We want to find c and p in $r_3(\varepsilon)^3 = \varepsilon + c\varepsilon^p$. Let $\varepsilon > 0$, and $r_\varepsilon = r_3(\varepsilon)$ be the corresponding pseudospectral radius. Using these values, we let $y = \log(r_\varepsilon - \varepsilon)$ and $x = \log(\varepsilon)$. Notice that this matches the form of equation (2.3), or $y = ax + b$ where $a = p$ and $b = \log(c)$. Record the values for x and y then repeat this process for a number of different ε values. Now we use MATLAB to fit a linear polynomial to our set of generated points. The coefficient, a, for x will be p in our formula and the intercept, b, will be $\log(c)$, or $e^b = c$. Running this process gives us the values $c = 1$ and $p = 5/3$. Below is a plot of the linear trend the calculated points for around 20 values of epsilon.



Using this algorithm on various sized Jordan blocks, we get the following radii:

J_k	Radius of $\sigma_\varepsilon(J_k)$ as $\varepsilon \to \infty$
J_2	$r_2(\varepsilon)^2 = \varepsilon + \varepsilon^2$
J_3	$r_3(\varepsilon)^3 = \varepsilon + \varepsilon^{5/3} + \frac{5}{3}\varepsilon^{7/3} + \frac{1}{3}\varepsilon^{9/3} + \mathcal{O}(\varepsilon^{11/3})$
J_4	$r_4(\varepsilon)^4 = \varepsilon + \varepsilon^{6/4} + \frac{6}{4}\varepsilon^{8/4} + \frac{21}{8}\varepsilon^{10/4} + \mathcal{O}(\varepsilon^{12/4})$
J_5	$r_5(\varepsilon)^5 = \varepsilon + \varepsilon^{7/5} + \frac{7}{5}\varepsilon^{9/5} + \frac{56}{25}\varepsilon^{11/5} + \mathcal{O}(\varepsilon^{13/5})$
J_6	$r_6(\varepsilon)^6 = \varepsilon + \varepsilon^{8/6} + \frac{8}{6}\varepsilon^{10/6} + 2\varepsilon^{12/6} + \mathcal{O}(\varepsilon^{14/6})$

This table summarizes many of the results from this paper. These numbers are new in the literature and have several interesting patterns.

Let us compare these numbers to the Better Lower Bound we found in 3.4. Recall, the bound says

$$r_N(\varepsilon)^N \geq \varepsilon(1+\varepsilon)^{N-1}.$$

Looking at the right hand side of the equation, we can expand using the binomial theorem and write

$$\begin{aligned}\varepsilon(1+\varepsilon)^{N-1} &= \varepsilon\left(\binom{N-1}{0}\varepsilon^0 + \binom{N-1}{1}\varepsilon^1 + \dots\right)\\ &= \varepsilon + (N-1)\varepsilon^2 + \dots\end{aligned}$$

Hence, the second term in epsilon in every expansion for this bound is ε^2. The numerical tests suggest that this exponent is too large; in our tests, we found that the second term in epsilon is $\varepsilon^{(N+2)/N}$ and $\frac{N+2}{N} < 2$ for all $N > 2$. Since we take ε to be small, the Better Lower Bound decays too quickly. From these numbers, we state the following proposition.

Proposition 4.1. *For an $N \times N$ Jordan block, J_N, the radius of the ε-pseudospectrum is described by the polynomial*

$$r_N(\varepsilon)^N = \sum_{i=0}^{k} c_i \varepsilon^{\frac{N+2i}{N}}$$

where $c_i, k \in \mathbb{R}$ and k is an integer.

Besides the exponents, there seems to be a pattern in the coefficients. The coefficient for the third term in the expansion is always $\frac{N+2}{N}$. Further, if we exclude the expansion for $r_3(\varepsilon)^3$, then the coefficient for the fourth term is $\frac{N+2}{N} \cdot \frac{N+3}{N}$. There appears to be structure contained within these expansions, waiting to be unlocked.

The big O notation indicates where the algorithm breaks down because of numerical precision issues. However, having the a few terms in each equation give us a target when finding classes of perturbations that get us close lower bounds on the pseudospectral radii. One thing that inspires hope in these numbers is that the algorithm does not return a valid number for the third term in the Puiseux series for the case where $N = 2$.

To prove this, we ideally want to come up with an error matrix or pseudoeigenvector for the Jordan blocks that gives us a pseudospectral radius that matches this kind of expansion. More work should be done figuring out the nature of these Puiseux series and the expansion coefficients.

4.2 Upper Bound on the Radius of the Jordan Block

So far, we have focused primarily on lower bounds for the pseudospectrum of the Jordan block, but now we will consider bounding the pseudospectrum from above. Right now, our best upper bound is the one established in Theorem 3.1. Since the Jordan block has symbol $f(z) = z$, the upper bound is

$$f(\Delta) + \Delta_\varepsilon = \Delta + \Delta_\varepsilon = \Delta_{1+\varepsilon}$$

or the ball centered at the origin with radius $1 + \varepsilon$. Clearly this is a bad upper bound since one dominates the radius for our small values of epsilon. We can use the results from the previous section to establish a better upper bound for the pseudospectra.

If Proposition 4.1 is correct, then we should be able to find an upper bound that depends on epsilon and contains the pseudospectra of our Jordan blocks. Namely, if $r_N(\varepsilon)^N = \varepsilon + \varepsilon^{(N+2)/N} + \mathcal{O}(\varepsilon^{(N+4)/N})$, then for small values of epsilon, we should have $r_N(\varepsilon)^N < \varepsilon + \varepsilon^{(N+1)/N}$. We explore this upper bound stochastically. In the plot below, we look at the 3×3 Jordan block. Two circles are drawn: the red one has radius $1 + \varepsilon$ and the blue has radius $\sqrt[N]{\varepsilon + \varepsilon^{(N+1)/N}}$. We then generate points in the pseudospectrum by creating error matrices E_i for $i = 1, 2, ...2000$ with $\|E_i\| \leq 0.1$. We then calculate the eigenvalues of $J_3 + E_i$ and plot them on the graph. We see that every resulting point is contained in within our blue circle.

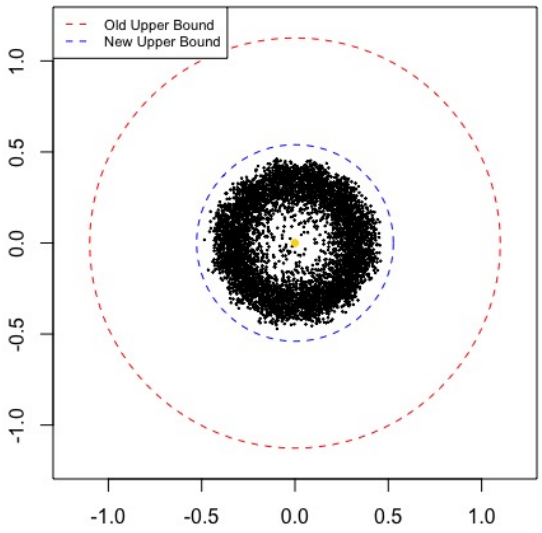


Figure 8: Upper Bounds for $\sigma_\varepsilon(J_3)$

A similar story is told when we vary the value of epsilon. The following plots show the same containment exists even when we take random values for epsilon. In each case the ball of radius $\sqrt[N]{\varepsilon + \varepsilon^{(N+1)/N}}$ contains all the points in our generated pseudospectrum.

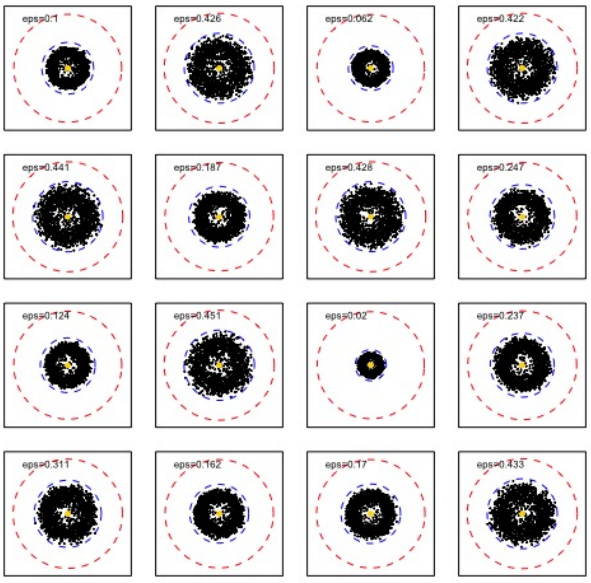


Figure 9: Upper Bounds for $\sigma_\varepsilon(J_3)$, Varying ε

And finally we have the analogous plots for the Jordan blocks of sizes $N =$

$4, 5, 6, 7$. Again, the blue circles contain the pseudospectra.

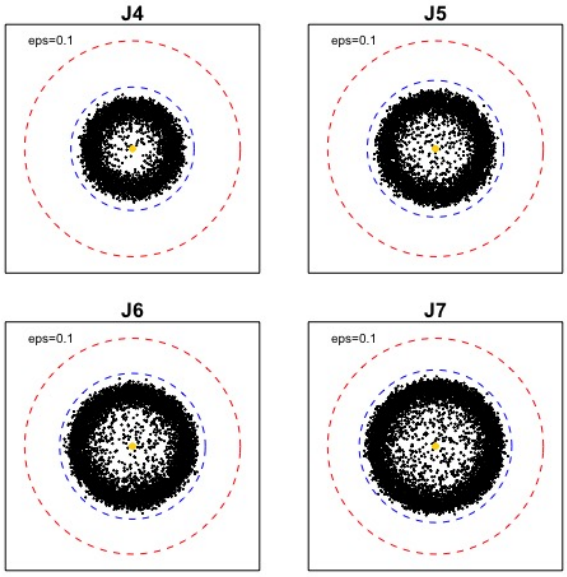


Figure 10: Upper Bounds for $\sigma_\varepsilon(J_N)$, Varying N

As another check, we plot a lower bound and our new upper bound on the same graph for the 3×3 Jordan block, J_3. For the upper bound, we still use $B(0, \sqrt[3]{\varepsilon + \varepsilon^{(N+1)/N}})$. For our lower bound, we take the first two terms of our proposed Puseiux series for the pseduospectral radius of J_3, giving us the lower bound $B(0, \sqrt[3]{\varepsilon + \varepsilon^{(N+2)/N}})$. In the following plot, we let $\varepsilon = 0.2$ and take 20,000 random perturbations. The points in the generated pseudospectrum with absolute value greater than the lower bound are plotted in red (although some appear to be contained in the lower bound because of graphical precision issues). Several values clearly break through this lower bound, but all are contained within our proposed upper bound.

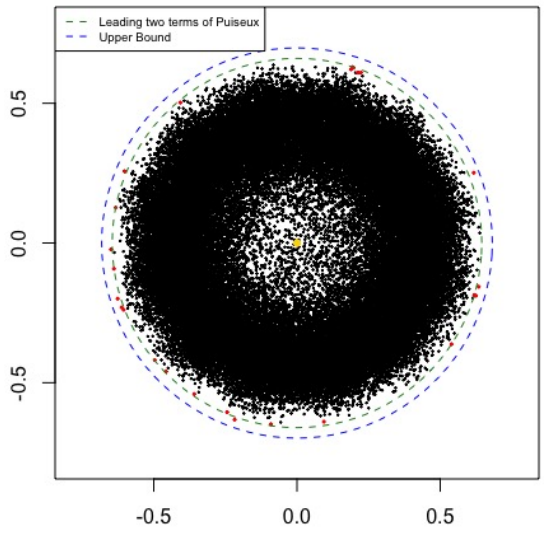


Figure 11: Lower and Upper Bounds for $\sigma_\varepsilon(J_3)$

This stochastic examination of the upper bound provides further evidence for 4.1, and leads us to another proposition.

Proposition 4.2. *Let J_N be an $N \times N$ Jordan block. Then*

$$\sigma_\varepsilon(J_N) \subseteq B(0, \sqrt[N]{\varepsilon + \varepsilon^{(N+1)/N}}).$$

This proposition is left as an area for future research. Of course, coming up with the exact radius is the ultimate goal. However, these propositions provide guidance for the types of upper and lower bounds we want to achieve, and numerical results to support the proposals. Hopefully these numbers provide a beacon for further research and thought.

5 References

[1] N. J. Higham, "Review of"spectra and pseudospectra: The behavior of nonnormal matrices and operators", by lloyd n. trefethen and mark embree. princeton university press, princeton, nj, usa, 2005.," *Bull. Amer. Math. Soc.*, vol. 44, no. 2, pp. 277–284, 2007.

[2] L. N. Trefethen and M. Embree, *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

[3] J. F. Grcar, "Matrix stretching for linear equations," *arXiv preprint arXiv:1203.2377*, 2012.

[4] C. Kuehn, "Introduction to potential theory via applications," *arXiv preprint arXiv:0804.4689*, 2008.

[5] T. Kato, *Perturbation theory for linear operators*, vol. 132. Springer Science & Business Media, 2013.

[6] A. Van der Sluis, "Condition numbers and equilibration of matrices," *Numerische Mathematik*, vol. 14, no. 1, pp. 14–23, 1969.

[7] F. Gong, O. Meyerson, J. Meza, M. Stoiciu, and A. Ward, "Explicit bounds for the pseudospectra of various classes of matrices and operators," *Involve, a Journal of Mathematics*, vol. 9, no. 3, pp. 517–540, 2016.

[8] F. L. Bauer and C. T. Fike, "Norms and exclusion theorems," *Numerische Mathematik*, vol. 2, no. 1, pp. 137–141, 1960.

[9] L. Reichel and L. N. Trefethen, "Eigenvalues and pseudo-eigenvalues of toeplitz matrices," *Linear algebra and its applications*, vol. 162, pp. 153–185, 1992.

[10] G. Beek, "Spectra of finite and infinite jordan blocks," Master's thesis, University of Twente, 2015.

[11] K. Knopp, *Theory of functions,*. New York: Dover publications, 1945.

[12] F. Aroca, G. Ilardi, and L. López de Medrano, "Puiseux power series solutions for systems of equations," *International Journal of Mathematics*, vol. 21, no. 11, pp. 1439–1459, 2010.

6 Appendix of Code

Note that Multiprecision Computing Toolbox for MATLAB (available at `advanpix.com`) is required to run these scripts.
To download the scripts, visit `https://github.com/abbank92/AppendixCode`

plotEigLevels.nb

```
(*Function for plotting pseudospectra level curves*)
(*(c) Alex Bank 2020*)
(*Parameters xmin, xmax, ymin, and ymax define the plot limits*)

plotEigsLevels[mat_, xmin_, xmax_, ymin_, ymax_] :=
 Module[{eigs, c, n},
  eigs = Eigenvalues[N[mat]];
  With[{c = Norm[mat], n = Length[mat]},
   ContourPlot[-c Log10[
      First[SingularValueList[
        mat - SparseArray[Band[{1, 1}] -> x + I y, {n, n}], -1,
        Method -> "Arnoldi", Tolerance -> 0]]], {x, xmin, xmax}, {y,
     ymin, ymax}, AspectRatio -> Automatic,
    Contours -> Range[1, 6, 1/2], ContourShading -> None,
    ContourStyle -> Black,
    Epilog -> {Directive[Black, AbsolutePointSize[5]],
      Point[ReIm /@ eigs]}]]]

(*Example usage using Grcar matrix with N=32*)
grcar[r : _Integer?Positive : 3, n_Integer?Positive] :=
 SparseArray[{{j_, k_} /; j == k + 1 :> -1, {j_, k_} /;
     0 <= k - j <= r :> 1}, {n, n}]

mat = grcar[32];
plotEigsLevels[mat, -2, 4, -4, 4]

(*The 4x4 Jordan Block*)
```

```
mat2 = {{0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}, {0, 0, 0, 0}};
plotEigsLevels[mat2, -1, 1, -1, 1]
```

numericalTestMP.m

This is the script that computes the Puiseux series expansion for each Jordan block

```
%% Function Test

% Set up a bunch of epsilon values
low = mp('10e7');
high = low*100;
inc = 990000;
smallSeq = low:inc:high;
eps = arrayfun(@(x) 1/x, smallSeq);

%% First Jacobi block... k = 3
k = 3;
J3 = mp([0 1 0; 0 0 1; 0 0 0]);

% Get corresponding radii
radEpsJ3 = mp(zeros(1, length(eps)));
for ind = 1:length(eps)
    radEpsJ3(ind) = findradiusMP(J3,eps(ind));
end

knownTerms = [1 1];
[p1j3, c1j3] = findNextPower(eps, radEpsJ3, k, knownTerms);
% Returns p1 = 1.666
%         c1 = 1

knownTerms = [1 1; 1 5/3];
[p2j3, c2j3] = findNextPower(eps, radEpsJ3, k, knownTerms);
% Returns p2 = 2.333
%         c2 = 1.666

knownTerms = [1 1; 1 5/3; 5/3 7/3];
[p3j3, c3j3] = findNextPower(eps, radEpsJ3, k, knownTerms);
```

```
% Returns p3 = 2.96 ~ 3 (which would be the next one in the series)
%         c3 = 0.178

%% Moving on... k=4
k = 4;
J4 = mp([0 1 0 0; 0 0 1 0; 0 0 0 1; 0 0 0 0]);

radEpsJ4 = mp(zeros(1, length(eps)));
for ind = 1:length(eps)
    radEpsJ4(ind) = findradiusMP(J4,eps(ind));
end

knownTerms = [1 1];
[p1j4, c1j4] = findNextPower(eps, radEpsJ4, k, knownTerms);
% p1 --> 1.5
% c1 --> 1

knownTerms = [1 1; 1 1.5];
[p2j4, c2j4] = findNextPower(eps, radEpsJ4, k, knownTerms);
% p2 --> 2
% c2 --> 1.5

knownTerms = [1 1; 1 1.5; 1.5 2];
[p3j4, c3j4] = findNextPower(eps, radEpsJ4, k, knownTerms);
% p3 --> 2.5
% c3 --> 2.625

knownTerms = [1 1; 1 1.5; 1.5 2; 2.625 2.5];
[p4j4, c4j4] = findNextPower(eps, radEpsJ4, k, knownTerms);

%% Moving on... k=5
k = 5;
J5 = mp([0 1 0 0 0; 0 0 1 0 0; 0 0 0 1 0; 0 0 0 0 1; 0 0 0 0 0]);
```

```
radEpsJ5 = mp(zeros(1, length(eps)));
for ind = 1:length(eps)
    radEpsJ5(ind) = findradiusMP(J5,eps(ind));
end

knownTerms = [1 1];
[p1j5, c1j5] = findNextPower(eps, radEpsJ5, k, knownTerms);
% p1 --> 1.4
% c1 --> 1

knownTerms = [1 1; 1 1.4];
[p2j5, c2j5] = findNextPower(eps, radEpsJ5, k, knownTerms);
% p2 --> 1.8
% c2 --> 1.4

knownTerms = [1 1; 1 1.4; 1.4 1.8];
[p3j5, c3j5] = findNextPower(eps, radEpsJ5, k, knownTerms);
% p3 --> 2.2
% c3 --> 2.24

k=5;
knownTerms = [1 1; 1 1.4; 1.4 1.8; 2.24 2.2];
[p4j5, c4j5] = findNextPower(eps, radEpsJ5, k, knownTerms);

%% Moving on... k=6
k = 6;
J6 = mp([0 1 0 0 0 0; 0 0 1 0 0 0; 0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1; 0 0 0 0 0

radEpsJ6 = mp(zeros(1, length(eps)));
for ind = 1:length(eps)
    radEpsJ6(ind) = findradiusMP(J6,eps(ind));
end

knownTerms = [1 1];
```

```
[p1j6, c1j6] = findNextPower(eps, radEpsJ6, k, knownTerms);
% p1 --> 1.333
% c1 --> 1

knownTerms = [1 1; 1 8/6];
[p2j6, c2j6] = findNextPower(eps, radEpsJ6, k, knownTerms);

knownTerms = [1 1; 1 8/6; 8/6 10/6];
[p3j6, c3j6] = findNextPower(eps, radEpsJ6, k, knownTerms);

k=6;
knownTerms = [1 1; 1 8/6; 8/6 10/6; 2 2];
[p4j6, c4j6] = findNextPower(eps, radEpsJ6, k, knownTerms);
```

findRadiusMP.m

```
function [f, z] = findradiusMP(A, epsln)
% Alex Bank MP update (January 17 2020)
% Michael Overton and Emre Mengi (Last Update on Spetember 2 2014)
% call: function [f, z] = pspr_2way(A, epsln)
%  Quadratically convergent two-way method to compute
%  the eps-pseudospectral radius of A.
%
%
%  The epsln-pseudospectral radius of A is the globally optimal value of:
%      max |z|
%      s.t. sigma_min(A - z I) = epsln     (smallest singular value)
%
%
%  input
%     A          square matrix
%     epsln      real >= 0 (0 implies f is spectral radius)
%
%  output
%     f          the epsln-pseudospectral radius of A
%     z          one of the global maximizers except when
%                the matrix is real. If the input matrix is real
%                it contains either a global optimizer on the
%                real axis or a complex conjugate pair.

% error checks
if isnaninf(A)|isnaninf(epsln)
   error('pspa_2way: nan or inf arguments not allowed')
end

if epsln < 0 | imag(epsln) ~= 0
   error('pspr_2way: epsln must be nonnegative real')
end
```

```
n = length(A);
if size(A) ~= [n n]
   error('A must be square')
end

if (n >= 30)
    h = waitbar(0,'Computing Pseudospectral Radius...(initializing)');
else
    h = 0;
end

% draw the eigenvalues of A
eA = eig(A);

smalltol = mp('1e-18')*max(norm(A),epsln);
purespr = max(abs(eA));     % pure spectral radius
if epsln == 0  % pseudospectrum is just the spectrum
   [sortreal, indx] = sort(-abs(eA));
   eA = eA(indx);
   f = purespr;

   % return all of the eigenvalues whose magnitudes are equal to spectral
   % radius
   ind = find(abs(eA) >= f - smalltol);
   z = eA(ind);

   if (n >= 30)
       waitbar(1,h,'Computing Pseudospectral Radius...(completed)')
   end

   if (n >= 30)
       close(h)
   end
else
```

```
rold = mp('-0.00001');
% initially r is spectral-radius
[r,ind] = max(abs(eA));  % initial iterate
% theta is the angles of the eigenvalues z, s.t |z| = r
% one such angle is sufficient, if just the pseudospectral radius
% is needed, but to return accurate global optimizers, all
% such angles should be computed.
theta = angle(eA(ind));
thetaold = theta;

thetabest = [];

iter = 0;
no_thetaeig = 0;
mE = -epsln*eye(mp(n));

% realtol is used to detect zero real parts up to some tolerance.
realtol = smalltol;
% radtol is the tolerance determining how far the eigenvalue magnitudes
% can be apart from unity (Used by the circular search).
radtol = smalltol;

while ~no_thetaeig & r > rold    % r is increasing in exact arithmetic

   iter = iter + 1;

   if iter > 20
       if (n >= 30)
         close(h)
       end
      error('pspr_2way: too many steps')
   end
```

```
thetabestt = thetabest;
rold = r;

% given the resulting directions in theta(computed in the
% previous iteration except when iter=1), look for the circle with
% the greatest radius intersecting pseudo-spectrum boundary.
% note: input theta is a vector, but output r is a scalar: the max
% of the max values.  thetabest is the corresponding theta value and is
% a scalar, even if there was a tie e.g. from a complex conjugate pair.
[r, thetabest] = radiushelperMP(A, mE, theta, realtol, rold, iter,h);

ptout = sprintf('Computing Pseudospectral Radius...(iteration %d)',iter);

if (n >= 30)
    waitbar((2*(iter-1)+1)/12,h,ptout)
end

if r > rold
 % given current r, look for all relevant intersections of the circle
 % with radius r with the pseudospectrum, process and return pair midpoints.
 % note: input r is a scalar, but output theta is a vector.
 % the input thetabest is a scalar.
 % there is one difference compared to pseudo-absicca case
 % let theta contains t1 <= t2 <= ...  <= tn
 % the intervals are circular that is we have to check the midpoints
 % of [t1,t2], [t2,t3], ..., [tn-1,tn],[tn,t1].
 % In one of these directions there should be a point on the
 % boundary with a greater radius compared to current maximum radius
  thetaold = theta;
 theta = findthetaMP(A, mE, epsln, r, thetabest, iter, radtol, ...
      smalltol, h);
  if (n >= 30)
      waitbar((2*(iter-1)+2)/12,h)
  end
```

```
    no_thetaeig = isempty(theta);

     if (no_thetaeig)
         rold = r;
     end

 else
     thetabest = thetabestt;
     r = rold;
 end % end of else
end % end of while

if (n >= 30)
    waitbar(1,h,'Computing Pseudospectral Radius...(completed)')
end

if isempty(thetabest)
     if (n >= 30)
         close(h)
     end
    error('Failed in the first iteration (please choose a bigger epsilon)');
end

% set f to the eps-pseudospectral radius
f = r;
z = r*(cos(thetabest) + mp('1i')*sin(thetabest));

if isreal(A) & ~isreal(z)
    z = [z; r*(cos(thetabest) - mp('1i')*sin(thetabest))];
end

if (n >= 30)
    close(h)
```

```
    end
end
```

radiusHelperMP.m

```
function [rbest, thetabest] = radiushelperMP(A, mE, theta, realtol, rold, iter,h)
% Alex Bank MP update (January 17 2020)
% Michael Overton and Emre Mengi (Last update on September 2 2014)
% called by pspr_2way.m
% Given a set of angles in theta, in each direction finds the point
% on the pseudospectrum boundary with the largest radius, i.e.
%           rnew(j) = max |z|
%                     s.t. sigma_min(A - z I) = epsln and angle(z) = theta(j)
% rbest is the maximum of the largest radius in any direction
%           rbest = max rnew(j)
%                 1<=j<=length(theta)
% thetabest is the angle of a point on the boudary with radius rbest in one of
% the directions in theta, i.e. let rbest = rnew(k), for some k, 1<=k<=length(theta
% then thetabest = theta(k).

n = length(A);
for j=1:length(theta)

K1 = [mp('1i')*exp(mp('1i')*theta(j))*transpose(A) mE];
    K2 = [-mE mp('1i')*exp(mp('-1i')*theta(j))*A];
    K = [K1;K2];
eK = eig(K);

   if min(abs(real(eK))) <= realtol % check if the matrix K has an imaginary eigenv
      indx = find(abs(real(eK)) <= realtol);  % extract such eigenvalues
      rnew(j) = real(max(imag(eK(indx))));

   else
      % there may be no point on the boundary of the pseudospectrum
      % in the direction theta(j)
      rnew(j) = mp('-inf');
   end % end of else
```

```
end % end of for

if isempty(rnew)
    if (n >= 30)
       close(h)
    end
    error('no intersection point is found by the radial search(please try smaller e
end

% choose the maximum of the largest radius in directions included in theta
[rbest,ind] = max(rnew);
thetabest = theta(ind);
```

findthetaMP.m

```
function thetanew = findthetaMP(A, mE, epsln, r, thetawant, iter, radtol, ...
     smalltol, h)
% Alex Bank MP update (January 17 2020)
% Michael Overton and Emre Mengi (Last Update on September 2 2014)
% called by pspr_2way.m
% Given a radius r, it first computes the intersection points
% of eps-pseudospectrum boundary with the circle with radius r.
% This is achieved by finding the generalized eigenvalues of the
% matrix pencil F - lambda*G where
%
%       F=[-eps*I A;r*I 0], G=[0 r*I; A' -eps*I]
%
% and then performing a singular value test. The singular value test
% is neccessary to eliminate the points for which eps is a singular
% value but not the smallest one. Finally the midpoints of two
% succesive intersection points on the circle with radius r is
% calculated, i.e. let re^(i*theta_i) and re^(i*theta_(i+1)) be the
% ith and (i+1)th intersection points, then ith midpoint is
% re^(i*(theta_i+theta_(i+1))/2). Special attention is paid to
% keep the angles in the interval [-pi,pi). Furthermore, as it
% was the case for pseudo-absicca code, we specifically considered
% the case when the angle from the previous iteration is contained
% in one of the intervals. At the exit thetanew contains the
% angles of the midpoints.

n = length(A);

% compute the generalized eigenvalues of the matrix pencil F - lambda*G
R = r * eye(mp(n));
O = 0 * eye(mp(n));
F = [mE A; R O];
G = [O R; A' mE];
eM = eig(F,G);
```

```
% extract the eigenvalues with magnitude 1
% a small tolerance is used
ind = find((abs(eM) < (mp('1') + radtol)) & (abs(eM) > (mp('1') - radtol)));
eM = eM(ind);


if (isempty(eM)) % check if M has an eigenvalue with magnitude 1
   thetanew = [];
else

   % sort eM wrt theta values
   [theta, indx] = sort(angle(eM));
   theta = angle(eM(indx));

   % perform singular value test on the points probably on
   % eps-pseudospectrum boundary.
   % the ones actually on the eps-pseudospectrum are those with smallest
   % singular value equal to eps
   indx2 = [];
   for j = 1: length(theta)
   if (theta(j) < 0)
theta(j) = theta(j) + mp('2*pi');
   end

       Ashift = A - (r*(cos(theta(j)) + mp('1i')*sin(theta(j))))*eye(mp(n));
       s = svd(Ashift);
       [minval,minind] = min(abs(s-epsln));

       if minind == n
           indx2 = [indx2; j];   % accept this eigenvalue
       end
   end
```

```
removed = length(theta) - length(indx2);

if removed > 0
    theta = theta(indx2);
end

if (isempty(theta))
     if (n >= 30)
         close(h)
     end
    error('singular value test removed all of the intersection points(please try
end

theta = sort(theta);

% organize in pairs and take midpoints
thetanew = [];
ind = 0;

% shift thetawant, the angle from the previous iteration, into
% the interval [0,2pi]
if (thetawant < 0)
    thetawant = thetawant + mp('2*pi');
end

for j=1:length(theta)
   thetalow = theta(j);

   if (j < length(theta))
       thetahigh = theta(j+1);
   else
% the last interval wraps around
       thetahigh = theta(1) + mp('2*pi');
   end
```

```
      % before taking the midpoint, if this interval is not very short,
      % check and see if thetawant is in this interval, well away from the
      % end points.  If so, break this pair into two pairs, one above
      % and one below thetawant, and take midpoints of both.
      inttol = mp('.01') * (thetahigh - thetalow);

  % this is needed for the last interval
  if (thetawant+mp('2*pi') > thetalow + inttol & ...
  thetawant+mp('2*pi') < thetahigh - inttol)
  thetawantt = thetawant + mp('2*pi');
  else
  thetawantt = thetawant;
  end

      if thetawantt > thetalow + inttol & ...
                thetawantt < thetahigh - inttol

         % lower midpoint
 thetamid = (thetalow + thetawantt)/mp('2');

 % shift thetamid into the interval [-pi,pi] again
 if (thetamid >= mp('2*pi'))
thetamid = thetamid - mp('2*pi');
 end
 if (thetamid >= pi)
thetamid = thetamid - mp('2*pi');
 end

 % remove the midpoint if the minimum singular value is greater than
 % epsilon, since in this case the midpoint should lie outside the
 % epsilon-pseudospectrum.
if (min(svd(A-r*exp(mp('1i')*thetamid)*eye(mp(n)))) <= epsln)
ind = ind + 1;
```

```
thetanew(ind,1) = thetamid;
end

         % upper midpoint
         thetamid = (thetawantt + thetahigh)/mp('2');

         % shift thetanew(ind) into the interval [-pi,pi] again
 if (thetamid >= mp('2*pi'))
thetamid = thetamid - mp('2*pi');
 end
 if (thetamid >= mp('pi'))
thetamid = thetamid - mp('2*pi');
 end
 % remove the midpoint if the minimum singular value is greater than
 % epsilon, since in this case the midpoint should lie outside the
 % epsilon-pseudospectrum.
 if (min(svd(A-r*exp(mp('1i')*thetamid)*eye(mp(n)))) <= epsln)
ind = ind + 1;
thetanew(ind,1) = thetamid;
 end

     else
         % otherwise, if thetawant is not in the interval
         % take the midpoint of thetalow and thetahigh
         thetamid = (thetalow + thetahigh)/mp('2');

 % shift thetanew(ind) into the interval [-pi,pi] again
 if (thetamid >= mp('2*pi'))
thetamid = thetamid - mp('2*pi');
 end
 if (thetamid >= mp('pi'))
thetamid = thetamid - mp('2*pi');
 end
```

```
 % remove the midpoint if the minimum singular value is greater than
 % epsilon, since in this case the midpoint should lie outside the
 % epsilon-pseudospectrum.
 if (min(svd(A-r*exp(mp('1i')*thetamid)*eye(mp(n)))) <= epsln)
ind = ind + 1;
thetanew(ind,1) = thetamid;
 end

      end
   end

  % if A is real, discard the midpoints in the lower half plane
  if isreal(A)
     indx = find(thetanew >= 0 | thetanew == mp('-pi'));
     ynew = thetanew(indx);
  end

end
```

isnaninf.m

```
function bval = isnaninf(A)
% call : bval = isnaninf(A)
%  returns 1 if A contains an inifinite or a nan entry, otherwise
%  returns 0

%  input
%     A          input matrix, vector or scalar
%  output
%     bval       1 if A has a infinite or a nan entry, 0 otherwise.


if isnan(A) | isinf(A)
    bval = 1;
else
    bval = 0;
end
```

upperBound.R

```
# upperBound.R
# (c) Alex Bank 2020

# Load library for drawing circles
library(plotrix)

# Function to generate Jordan blocks
genJ <- function(N) {
  J <- matrix(rep(0,N*N),nrow = N,ncol = N)
  for (i in 1:(N-1)) J[i,i+1] <- 1
  return(J)
}

# Function for generating a complex number
rcomp <- function(x){
  return(complex(
    real = runif(1,-1,1),
    imaginary = runif(1,-1,1)))
}

# Function for generating an error matrix
# with norm <= eps
genE <- function(eps,N){
  errM <- matrix(
    sapply(rep(0,N*N),rcomp),
    nrow = N,ncol = N)
  m <- norm(errM,type='2')
  if (m>eps) {
    x <- eps/norm(errM,type = "2")}
  else {
    x <- 1}
  return(x*errM)
}
```

```
# Function for returning eigenvalues
# of A+E
genPseudo <- function(A,E){
  return(eigen(A+E)[[1]])
}

# Function for generating the plots
genPlot <- function(A,eps,ntimes,ax=TRUE,toplab=''){
  N <- dim(A)[1]
  eigList <- lapply(rep(0,ntimes),
                    function(x)
                      {return(genPseudo(A,genE(eps,N)))})
  origEigs <- eigen(A)[[1]]
  lims <- 1+eps+0.1
  if (ax) {
    plot(1,type="n", xlab="", ylab="",
         xlim=c(-lims, lims), ylim=c(-lims, lims),main = toplab)}
  else {
    plot(1,type="n", xlab="", ylab="",
         xlim=c(-lims, lims), ylim=c(-lims, lims),
         xaxt='n',yaxt='n',main=toplab)
  }
  for (e in eigList) {
    points(Re(e),Im(e),pch=20,cex=.25)
  }
  points(Re(origEigs),Im(origEigs),col = 'gold',pch=20)
  draw.circle(0,0,1+eps,border='red',lty=2)
  draw.circle(0,0,(eps+eps^((N+1)/N))^(1/N),
              border='blue',lty=2)
  if (ax) {
  legend("topleft",legend = c("Old Upper Bound","New Upper Bound"),
         col=c('red','blue'),lty=2,cex=.7)}
  else {
```

```
    legend("topleft", legend = paste0("eps=",round(eps,3)),cex = .75,bty='n')
  }
}

# Figure 1
par(mai =rep(.5,4))
J3 <- genJ(3)
genPlot(J3,.1,2000)

# Figure 2
par(mfrow = c(4,4), mai = rep(.1,4))
genPlot(J3,.1,500,FALSE)
for (i in 1:15) genPlot(J3,runif(1,0,.5),500,FALSE)

# Figure 3
par(mfrow = c(2,2), mai = rep(.2,4))
J4 <- genJ(4); J5 <- genJ(5); J6 <- genJ(6); J7 <- genJ(7)
genPlot(J4,.1,1000,FALSE,"J4")
genPlot(J5,.1,1000,FALSE,"J5")
genPlot(J6,.1,1000,FALSE,"J6")
genPlot(J7,.1,1000,FALSE,"J7")
```

upperVlowerBound.R

```
# upperVlowerBound.R
# (c) Alex Bank 2020

genPlot2 <- function(A,eps,ntimes,ax=TRUE,toplab=''){
  N <- dim(A)[1]
  eigList <- lapply(rep(0,ntimes),
                    function(x)
                    {return(genPseudo(A,genE(eps,N)))})
  origEigs <- eigen(A)[[1]]
  lims <- (eps+eps^((N+1)/N))^(1/N)+0.1
  lowerB <- (eps+eps^((N+2)/N))^(1/N)
  if (ax) {
    plot(1,type="n", xlab="", ylab="",
         xlim=c(-lims, lims), ylim=c(-lims, lims),main = toplab)}
  else {
    plot(1,type="n", xlab="", ylab="",
         xlim=c(-lims, lims), ylim=c(-lims, lims),
         xaxt='n',yaxt='n',main=toplab)
  }
  for (e in eigList) {
    for(indEig in e) {
      if (abs(indEig) > lowerB){
        points(Re(indEig),Im(indEig),pch=20,cex=.4,col='red')
      }
      else {points(Re(indEig),Im(indEig),pch=20,cex=.25)}
    }
  }
  points(Re(origEigs),Im(origEigs),col = 'gold',pch=20)
  draw.circle(0,0,(eps+eps^((N+2)/N))^(1/N),border='forestgreen',lty=2)
  draw.circle(0,0,(eps+eps^((N+1)/N))^(1/N),
              border='blue',lty=2)
  if (ax) {
    legend("topleft",legend = c("Leading two terms of Puiseux","Upper Bound"),
```

```
          col=c('forestgreen','blue'),lty=2,cex=.6)}
  else {
    legend("topleft", legend = paste0("eps=",round(eps,3)),cex = .75,bty='n')
  }
}

genPlot2(J3,.2,20000)
```