ON MATRIX NEARNESS PROBLEMS: DISTANCE TO DELOCALIZATION

VLADIMIR R. KOSTIĆ*, AGNIESZKA MIĘDLAR†, AND JEROEN J. STOLWIJK†

Abstract. This paper introduces two new matrix nearness problems that are intended to generalize the distance to instability and the distance to stability. They are named the *distance to delocalization* and the *distance to localization* due to their applicability in analyzing the robustness of eigenvalues with respect to arbitrary localization sets (domains) in the complex plane. For the open left-half plane or the unit circle, the distance to the nearest unstable/stable matrix is obtained as a special case. Then, following the theoretical framework of Hermitian functions and the Lyapunov-type localization approach, we present a new Newton-type algorithm for the distance to delocalization (D2D) and study its implementations using both an explicit and an implicit computation of the desired singular values. Since our investigations are motivated by several practical applications, we will illustrate our approach on some of them. Furthermore, in the special case when the distance to delocalization becomes the distance to instability, we will validate our algorithms against the state of the art computational method.

Key words. stability, matrix nearness problems, distance to instability, distance to stability, spectral abscissa, stability radius, ε-pseudospectra, Lyapunov stability test, Hermitian functions, Newton's method

AMS subject classifications. 65F15, 15A18, 15A22

1. Introduction.

Numerous problems in mechanics, mathematical physics, and engineering can be formulated as eigenvalue problems where the focus is to determine whether the eigenvalues are inside a specific desirable domain [9, 18, 25, 30], and later on to detect an admissible size of a perturbation which will not move the eigenvalues away from that domain. The most frequent of such domains in use are connected to the stability of dynamical systems: the open left half-plane of the complex plane (continuous dynamical systems) and the open unit disk (discrete dynamical systems). Due to extensive research on the robustness of eigenvalues in the past years, it is now well known that the concept of pseudospectra [32] is a more adequate tool to analyze the behavior of nonnormal matrices, than the classical spectra. Because of this fact, many authors have considered quantities such as pseudospectral abscissa, pseudospectral radius, and in particular the distance to instability (stability), see [1, 6, 7, 10, 13, 14, 15, 16, 17, 21, 26, 27], in order to provide efficient tools for exploring phenomena of spectral change under small perturbations.

While the distance to instability and the distance to stability have attracted a lot of attention, there have not been many attempts to solve the same problem for some other regions in the complex plane, although such problems often arise in applications. For example, in eigenvalue problems from acoustic field computations, one is often not interested in all the eigenvalues in the right half plane, but is interested in only those whose imaginary part is between certain bounds defined by the hearing range of the human ear [29]. Therefore, the information about the size of the perturbation that will move the eigenvalues that are not originally in the hearing range (given by an unbounded rectangular area in the complex plane) into it, is of importance, since this would result in an undesirable sound, see [28]. Apart from this, many other

*Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg D. Obradovica 4 21000 Novi Sad, Serbia. vkostic@dmi.uns.ac.rs

†Institut für Mathematik, MA 4-5, TU Berlin, Strasse des 17. Juni 136, D-10623 Berlin, Germany. miedlar@math.tu-berlin.de, stolwijk@math.tu-berlin.de

problems in physics, structural mechanics and engineering and numerical analysis involve different spectral domains of interest. Motivated by this, we present Hermitian functions as a useful approach to treat different algebraic domains in the complex plane [24] and we formulate the nearness problems connected to such domains.

The paper is organized as follows. We introduce as a new concept a distance to delocalization (localization), motivate our work and briefly discuss the distance to instability in Section 2. In Section 3 we introduce the theoretical framework of the Lyapunov-type localization approach which we use to derive a computational method for the distance to delocalization in Section 4. For a large class of algebraic domains, we provide two numerical algorithms that solve the introduced distance problem under certain conditions. More precisely, we provide numerical procedures that compute the size of the Euclidian norm of a minimal unstructured perturbation that will move eigenvalues out of the desired region. Since the main aim is to develop techniques that can tackle large scale problems, the developed algorithms avoid frequent eigendecompositions and/or singular value decompositions [12, Chapter 5,7,8], and are mainly based on linear solvers for large scale linear systems, while the used nonlinear optimization approach is the implicit determinant method of [10] coupled with a Newton-like method. Finally, the last section contains numerical experiments for some well-known matrices with sensitive eigenvalues, comparisons with previously proposed methods and some interesting applications of the introduced approach.

Throughout this paper, matrices are square and complex of size n and are denoted by capital letters. The identity matrix is denoted by I and its dimensions follow from the context. A Hermitian matrix $X \in \mathbb{C}^{n,n}$ is called positive definite, denoted $X > 0$, if $v^*Xv > 0$ for all $v \in \mathbb{C}^{n,n}\backslash\{0\}$. Given a matrix $A \in \mathbb{C}^{n,n}$, $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_n(A)$ denote its singular values.

2. Preliminaries and motivations. Among many matrix nearness problems, e.g., distance to symmetry, positive approximation, distance to orthogonality, distance to normality or nearest matrix of low rank [19], the most commonly studied class involves the set of c-stable matrices, i.e., the set of matrices which eigenvalues are in the open left-half plane of the complex plane. Let us denote the set of c-stable matrices of size n by $\mathbb{S}^{n,n}$, i.e., for $A \in \mathbb{C}^{n,n}$, $A \in \mathbb{S}^{n,n}$ if and only if for every $\lambda \in \Lambda(A)$, $\mathrm{Re}(\lambda) < 0$, where $\Lambda(A) := \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\}$ denotes the spectrum of the matrix A. Obviously, $\mathbb{S}^{n,n}$ is a subspace of a normed space $(\mathbb{C}^{n,n}, \|\cdot\|)$, where the intended norm on $\mathbb{C}^{n,n}$ will be specified.
We first introduce the well-known problem of *distance to instability*:

Given a matrix $A \in \mathbb{S}^{n,n}$, determine $\widehat{A} \notin \mathbb{S}^{n,n}$, such that

$$\|\widehat{A} - A\| = \inf_{X \notin \mathbb{S}^{n,n}} \|X - A\|,$$

which can be considered as a measure of the robustness of matrix stability under arbitrary perturbations. A different, complementary problem is called *distance to stability*:

Given $A \notin \mathbb{S}^{n,n}$, determine $\widehat{A} \in \mathbb{S}^{n,n}$, such that

$$\|\widehat{A} - A\| = \inf_{X \in \mathbb{S}^{n,n}} \|X - A\|,$$

which can be viewed as the "stabilization" method for continuous dynamical systems.

In both cases, we are interested in finding a perturbation of the smallest norm such that the stability property will be either lost or gained. Therefore, we can formulate

both problems in terms of eigenvalue perturbation, namely, for a given $A \in \mathbb{C}^{n,n}$, one can compute the distance to instability $\delta^-_{stab} \geq 0$, and distance to stability $\delta^+_{stab} \geq 0$ using the following expressions:

$$\delta^-_{stab}(A) := \inf\{\|\Delta\| : \Lambda(A+\Delta) \not\subseteq \mathcal{LHP},\ \Delta \in \mathbb{C}^{n,n}\}, \tag{2.1}$$

$$\delta^+_{stab}(A) := \inf\{\|\Delta\| : \Lambda(A+\Delta) \subseteq \mathcal{LHP},\ \Delta \in \mathbb{C}^{n,n}\}, \tag{2.2}$$

where $\mathcal{LHP} := \{\lambda \in \mathbb{C} : \mathrm{Re}(\lambda) < 0\}$ denotes the open left-half plane of the complex plane.

Investigating the behavior of a general matrix under arbitrary perturbations Δ of norm $\|\Delta\| \leq \varepsilon$, with $\varepsilon > 0$, leads to the study of the *pseudospectra* [32]:

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : \|(A - zI)^{-1}\|^{-1} < \varepsilon\}.$$

More precisely, $\Lambda_\varepsilon(A)$ is called the ε-*pseudospectrum* of the matrix A in the norm $\|\cdot\|$. Its role becomes especially important for nonnormal matrices, i.e., when $AA^* \neq A^*A$. In this case, the size and the geometry of the pseudospectrum illustrate the dynamical behavior, while the eigenvalues itself do not provide this information, see [32, Part V, §23]. Particularly, for $\|\cdot\|$ being the Euclidian norm, denoted by $\|\cdot\|_2$,

$$\Lambda_\varepsilon(A) = \{z \in \mathbb{C} : \|(A - zI)^{-1}\|_2^{-1} < \varepsilon\} = \{z \in \mathbb{C} : \sigma_n(A - zI) < \varepsilon\},$$

where $\sigma_n(A)$ is the minimal singular value of the matrix A. As a consequence, one can determine the distance to instability $\delta^-_{stab}(A)$ using the pseudospectral theory by computing

$$\delta^-_{stab}(A) = \inf\{\varepsilon \geq 0 : \Lambda_\varepsilon(A) \not\subseteq \mathcal{LHP}\},$$

which is equivalent to

$$\delta^-_{stab}(A) = \min_{t \in \mathbb{R}} \sigma_n(A - \imath t I). \tag{2.3}$$

In practice, we encounter the following constrained optimization problem:

$$\begin{array}{l} \textit{Find } \min \varepsilon > 0 \quad \textit{s.t.} \quad (A - \imath t I)v = \varepsilon u \ \ \textit{and} \ \ (A - \imath t I)^* u = \varepsilon v, \\ \textit{where} \quad u, v \in \mathbb{C}^n, t \in \mathbb{R}. \end{array} \tag{2.4}$$

If the global minimum of (2.4) is obtained in $(\widehat{\varepsilon}, \widehat{u}, \widehat{v}, \widehat{t})$, then the distance to instability $\delta^-_{stab}(A)$ is given by $\delta^-_{stab}(A) = \widehat{\varepsilon}$ and the perturbation Δ that moves an eigenvalue of A to the point $\widehat{z} = \imath\widehat{t}$ on the imaginary axis is defined by $\Delta = -\widehat{\varepsilon}\widehat{u}\widehat{v}^*$. In addition, notice that $(\widehat{\varepsilon}, \widehat{u}, \widehat{v})$ defines the singular triplet of $(A - \imath\widehat{t}I)$.

Several authors have dealt with this problem. Let us briefly review some of the previous results. Following the early work [33], the SVD approach (computing the smallest singular value as in (2.3)) was researched intensively. In [7], a bisection method was proposed to determine the lower and upper bound on the distance to instability. One of the inconveniences in this approach is the necessity of solving a sequence of eigenvalue problems involving a large Hamiltonian matrix. This method provided the basis for [17], where the inverse iteration method for singular values was used to find a stationary point $S(t) := \sigma_n(A - \imath t I)$. The global minimum condition was checked afterwards by solving the Hamiltonian eigenvalue problem. More efficient approaches include a so-called criss-cross algorithm [6] and the extension of [7] in [26].

Recently, the so-called *implicit determinant method* was introduced in [1, 10]. The initial idea from [31] was extended to a Newton-based method for the calculation of the two-dimensional Jordan block corresponding to a purely imaginary eigenvalue in a two-parameter dependent Hamiltonian eigenvalue problem introduced in [7]. Since effective computations of the minimal singular triplets is challenging, the limits of the method concern large (and often sparse) matrices that one encounters in practice.

An alternative approach is based on locating zeros of the pseudospectal abcissa or radius function (the maximal real part or the maximal modulus of points in the ε-pseudospectrum). Using the fact that the computation of the pseudospectrum can be restricted to rank one perturbations, an iterative method for computing the ε-pseudspectral abscissa or radius of discrete dynamical systems was recently proposed in [16, 21]. For continuous dynamical systems an interesting idea of using differential equations on the manifold of normalized matrices of rank one was presented and successfully developed in [13, 14, 15].

The pseudospectral approach, though very suitable for computing the distance to instability, turned out to be improper for the complementary problem (distance to stability). In general, when the ε-pseudospectrum of the matrix crosses the imaginary axis, there is no method to verify if *all* eigenvalues that originated from the open left-half plane are still contained in this region (didn't leave this region). Due to this difficulty, much less attention was given to the problem of the distance to stability. Recently, an interesting method using a Lyapunov stability test has been proposed in [27]. In this approach, the distance to stability is treated in the Frobenius norm, and it is again formulated as a constrained optimization method, but instead of using the pseudospectrum, the constraints are given through positive definiteness of certain matrices. Namely, it is a well known fact, called *Lyapunov stability test* [23], that $A \in \mathbb{S}^{n,n}$ if and only if there exists an Hermitian positive definite (HPD) matrix Y such that $-(AY + YA^*)$ is also positive definite. Using this result, a computational method using successive convex approximations by Dikin ellipsoids [8] was introduced to compute the distance to stability defined in the following way:

$$\inf_{X,Y} \tfrac{1}{2}\|X - A\|_F \quad \text{s.t.} \quad -(XY + YX^*) > 0, \quad Y = Y^* > 0. \tag{2.5}$$

For $\widehat{X}$ being a global solution of (2.5), the distance to stability $\delta^+_{stab}(A)$ is given as $\delta^+_{stab}(A) = \|\widehat{X} - A\|_F$. Unfortunately, in order to obtain a Dikin ellipsoid which is as close as possible to the original nonconvex domain, the choice of a suitable Y and X is crucial for the quality of the method.

As we have seen, both problems, the distance to instability and distance to stability, are of particular importance from the practical point of view. However, there are many applications when one is interested to investigate the behavior of eigenvalues under perturbations beyond left-half plane or unit circle domains. For example, in structural acoustics, the localization of the eigenvalues in the complex plane corresponds to the appearance of acoustic waves of certain frequencies, [28]. In general, the range of frequencies that can be heard by humans is from 20Hz to 20kHz. This means that the eigenvalues in the left half-plane (stable modes) or those whose imaginary part, $\mathrm{Im}(\lambda)$, does not belong to the horizontal strips $(-2\pi \cdot 20kHz, -2\pi \cdot 20Hz)$ and $(2\pi \cdot 20Hz, 2\pi \cdot 20kHz)$ in the right-half plane (hearing range of unstable modes), correspond to "safe" modes in the sense that they do not produce audible noise. In the context of linear discrete dynamical systems, if the spectrum of the governing matrix belongs to the so-called *annulus* domain, i.e., $r < |z| < R$ with radii $R > r > 0$,

see Figure 5.2, then the dynamical system is stable and the dynamics matrix is non-singular due to the fact that r is the lower bound for the moduli of its eigenvalues. Also, in some applications, having the eigenvalues in the open left half-plane is often undesirable for the simple reason that stability may be a too weak constraint. Often it is desired that the spectrum guarantees that the corresponding closed loop dynamics is damped, [18], implying that the domain of interest could be a wedge around the negative part of the real line. One of such cases is the analysis of the bending vibrations of the airframe of a rocket where the domain of interest is the *cissoid of Diocles*, see Exmaple 4 and Figure 5.1.

Already these few examples certainly indicate the relevance of generalizing the distance to instability and the distance to stability to what we name here as the *distance to delocalization* and the *distance to localization*.

DEFINITION 2.1. *Let Ω be an open set in the complex plain, such that its boundary consists of piecewise continuous algebraic curves, then*

$$\inf\{\|\Delta\| : \Lambda(A+\Delta) \not\subseteq \Omega,\ \Delta \in \mathbb{C}^{n,n}\}$$

is called the ***distance to delocalization****, while*

$$\inf\{\|\Delta\| : \Lambda(A+\Delta) \subseteq \Omega,\ \Delta \in \mathbb{C}^{n,n}\}$$

is called the ***distance to localization****.*

Roughly speaking, the distance to delocalization quantifies the robustness of eigenvalue localization sets, whereas the distance to localization enables to construct a closest nearby matrix with its spectrum in a prescribed domain.

3. Distance to delocalization/localization. Before we generalize the distance to instability and distance to stability to other domains in the complex plane, we will shortly mention two major approaches for obtaining eigenvalue localization sets. The first one consists of forming unions and/or intersections of sets in the complex plane (disks, ovals, lemniscates, etc.) which are constructed for a given matrix. We can think of such localization sets as Geršgorin-type localizations, eg. the union of Geršgorin disks [11], see also [20, 34]. Another class of localization sets is defined through different domains in the complex plane (half-plane, unit disk, etc.) that are independent of the matrix itself. For each such domain a specific matrix test can be assigned, which verifies if the domain contains the eigenvalues of the matrix of interest. A well known localization test is given by the Lyapunov stability theorem [23]. Namely, if for a given matrix $A \in \mathbb{C}^{n,n}$, we define the operator $\mathcal{L}_A(X) = -(AX + XA^*)$, then the open left half-plane localizes eigenvalues of A if and only if $\mathcal{L}_A : \mathbb{H}^{n,n} \to \mathbb{H}^{n,n}$ is bijective, where $\mathbb{H}^{n,n}$ denotes the set of all n-by-n Hermitian positive definite matrices. We will now extend Lyapunov theorem as in [24], and refer to this class of localizations as Lyapunov-type localizations.

Here, we assume that the domain of the possible location of the spectrum is given. Then one is interested in the conditions of the desired inclusion of the eigenvalues with respect to the domain boundary. The main results in this direction are related to the computation of inertia indices of Hermitian solutions of linear matrix equations (generalizing the Lyapunov equations), using Hermitian functions with separable variables. In particular, given a Hermitian matrix $\Gamma = \Gamma^* = [\gamma_{pq}] \in \mathbb{C}^{m,m}$, $m \geq 2$, we will consider functions of the form

$$f(z) := \sum_{p=1}^{m}\sum_{q=1}^{m} \gamma_{pq}\varphi_p(z)\overline{\varphi_q(z)} = \varphi(z)^T \Gamma\, \overline{\varphi(z)}, \tag{3.1}$$

where $\varphi(z) = [\varphi_1(z), \varphi_2(z), \ldots, \varphi_m(z)]^T$ and the complex functions $\{\varphi_p\}_{p=1}^m$ are linearly independent. Since $\Gamma = \Gamma^*$, f is a real valued function of a complex variable and we can consider it as a map $f : \mathbb{R}^2 \to \mathbb{R}$. For notational convenience, in the remainder of the paper, we will identify $f(x + \imath y)$ with $f(x, y)$.
Let us now define the following sets:

$$\Lambda_f^+ := \{z \in \mathbb{C} : f(z) > 0\}, \tag{3.2}$$

$$\Lambda_f^- := \{z \in \mathbb{C} : f(z) < 0\}, \tag{3.3}$$

$$\Lambda_f^0 := \{z \in \mathbb{C} : f(z) = 0\}. \tag{3.4}$$

The domain in (3.4) can be considered as a curve in the complex plane separating the two other domains in (3.2) and (3.3). In general, however, the boundaries $\partial\Lambda_f^+$ and $\partial\Lambda_f^-$ of the domains in (3.2) and (3.3), respectively, do not necessarily have to coincide with Λ_f^0. However, if we choose the basis $\{\varphi_p\}_{p=1}^m$ as a standard basis, i.e., $\varphi_p(z) := z^{p-1}$, $1 \le p \le m$, then Λ_f^0 describes an algebraic curve of order at most $2m - 2$, and the inclusions $\partial\Lambda_f^+ \subseteq \Lambda_f^0$ and $\partial\Lambda_f^- \subseteq \Lambda_f^0$ hold.

Given a matrix $A \in \mathbb{C}^{n,n}$, a Hermitian function f defined as in (3.1) and basis functions $\{\varphi_p\}_{p=1}^m$, we construct the generalized Lyapunov operator $\mathcal{L}_A^f : \mathbb{C}^{n,n} \to \mathbb{C}^{n,n}$, given by

$$\mathcal{L}_A^f(X) := \sum_{p=1}^m \sum_{q=1}^m \gamma_{pq} \varphi_p(A) X \varphi_q(A^*), \tag{3.5}$$

where, for $1 \le p \le m$, $\varphi_p(A)$ is a matrix function defined via

$$\varphi_p(A) = -\frac{1}{2\pi\imath} \oint_{\mathfrak{C}} \varphi_p(z)(A - zI)^{-1} dz,$$

and $\mathfrak{C}$ is a closed simple contour separating a closed domain in $\mathbb{C}$ which contains the spectrum of A that excludes the singularities of φ_p.

In particular, for the standard basis, f can be written as

$$f(z) := \sum_{p=1}^m \sum_{q=1}^m \gamma_{pq} (z)^{p-1} (\overline{z})^{q-1}, \tag{3.6}$$

and therefore, since $\varphi_p(A) = A^{p-1}$, $1 \le p \le m$, the operator (3.5) takes the form

$$\mathcal{L}_A^f(X) := \sum_{p=1}^m \sum_{q=1}^m \gamma_{pq} A^{p-1} X A^{*(q-1)}.$$

In the following, for the sake of clarity, we develop algorithms for computing the distance to delocalization working with domains which can be expressed in the standard basis. The use of other bases will be illustrated later in several numerical examples. Further details about these domains and their properties, together with a method for computing their unions and intersections can be found in [24, Section 1.1. and 1.4]. Here, we just note that, introducing matrices

$$\Gamma_l = \begin{bmatrix} 2a\sin\theta & -\sin\theta + \imath\cos\theta \\ -\sin\theta - \imath\cos\theta & 0 \end{bmatrix}, \quad \Gamma_c = \begin{bmatrix} r^2 - |\omega|^2 & \omega \\ \overline{\omega} & -1 \end{bmatrix}, \tag{3.7}$$

$$\Gamma_s = \begin{bmatrix} 4a^2 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \text{ and } \Gamma_a = \begin{bmatrix} -r^2R^2 & 0 & 0 \\ 0 & r^2+R^2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \tag{3.8}$$

a half-plane whose boundary is a line $y\cos\theta = (x-a)\sin\theta$, $a \in \mathbb{R}$, $\theta \in [0,\pi)$, can be obtained using a matrix Γ_l in (3.7), the interior of a circle centered in $\omega \in \mathbb{C}$ with radius $r > 0$ using a matrix Γ_c in (3.7), a horizontal strip $y^2 = a^2$, $a > 0$, using matrix Γ_s in (3.8), and an annulus of radii $0 < r < R = 1$ centered at zero using matrix Γ_a in (3.8).

Our motivation to use Lyapunov-type domains arises from the following theorem.

THEOREM 3.1 ([22, 24]). *Let f be a Hermitian function given by (3.6) such that the Hermitian matrix $\Gamma\varphi(\overline{z})\varphi(z)^T\Gamma - f(z)\Gamma$ is positive semidefinite. Given a matrix $A \in \mathbb{C}^{n,n}$, and an arbitrary Hermitian positive definite matrix $Y \in \mathbb{C}^{n,n}$, all the eigenvalues of matrix A belong to the domain Λ_f^+ if and only if the equation $\mathcal{L}_A^f(X) = Y$ has a unique positive definite solution X, i.e.,*

$$\Lambda(A) \subseteq \Lambda_f^+ \text{ if and only if } \mathcal{L}_A^f : \mathbb{H}^{n,n} \to \mathbb{H}^{n,n} \text{ is a bijection.}$$

In fact, a more general Theorem holds, [24, Theorem 1.5.1]. For a certain class of Hermitian functions f, the number of eigenvalues of the matrix A inside the domains Λ_f^+ and Λ_f^- coincides with the number of positive and negative eigenvalues of the matrix X, respectively, where X is such that $\mathcal{L}_A^f(X)$ is positive definite. These Lyapunov-type localization techniques and inertia tests will play an essential role in studying the distance to localization, which is beyond the scope of this paper and will be the subject of future work.

So far, for a given $\Gamma = \Gamma^*$ and a basis $\{\varphi_p\}_{p=1}^m$, using Theorem 3.1, we can verify if the inclusion $\Lambda(A) \subseteq \Lambda_f^+(A)$ holds. Now, our main goal is to determine whether the ε-pseudospectrum of A is enclosed in the domain and to calculate the largest value of ε for which it is. This leads to the formulation of a new matrix nearness problem, called the *distance to delocalization* of $\Lambda(A)$ from the domain $\Lambda_f^+(A)$, i.e.,

$$\delta_f^-(A) := \inf\left\{\|\Delta\| : \Lambda(A+\Delta) \not\subseteq \Lambda_f^+,\ \Delta \in \mathbb{C}^{n,n}\right\}.$$

Obviously, since for $\Gamma = \Gamma_l$, $a = 0$ and $\theta = \frac{\pi}{2}$, Λ_f^+ becomes an open left-half plane, and for $\Gamma = \Gamma_c$, $\omega = 0$ and $r = 1$, Λ_f^+ becomes an open unit disk, the distance to delocalization is a generalization of the distance to instability in both the discrete and the continuous case.

As we have already noticed, a straightforward approach to solve this problem is using pseudospectral sets, where

$$\delta_f^-(A) = \sup\left\{\varepsilon : \Lambda_\varepsilon(A) \subseteq \Lambda_f^+\right\}.$$

Assuming that $\partial\Lambda_f^+ = \partial\Lambda_f^- = \Lambda_f^0$ and employing the Euclidean norm, yields

$$\delta_f^-(A) = \min_{z\in\Lambda_f^0} \sigma_n(A - zI). \tag{3.9}$$

Before we discuss a computational method to compute the distance to delocalization, which is the main objective of this paper, we will introduce a complementary

matrix nearness problem generalizing the distance to instability in the discrete and continuous sense. We define the *distance to localization* of $\Lambda(A)$ to the domain $\Lambda_f^+(A)$ as

$$\delta_f^+(A) := \inf\left\{\|\Delta\| : \Lambda(A+\Delta) \subseteq \Lambda_f^+,\ \Delta \in \mathbb{C}^{n,n}\right\}.$$

Again, Theorem 3.1 allows to express $\delta_f^+(A)$ equivalently as

$$\delta_f^+(A) = \inf\left\{\|\widehat{A} - A\| : \mathcal{L}_{\widehat{A}}^f(X) > 0,\ X > 0,\ \widehat{A} \in \mathbb{C}^{n,n}\right\}.$$

This formulation of the problem constitutes a starting point for generalizing the successive convex approximations for the stabilization of polynomials and matrices introduced in [27] into a computational method for the distance to localization.

4. Computational methods for distance to delocalization. In this section we develop computational methods for the distance to delocalization problems stated in (3.9). A singular value minimization problem over a curve in the complex plane is considered. Under the assumption that the solution is a simple singular value, we can construct a Newton-type method using formulas for the first and the second order derivatives of a (simple) minimal singular value. More precisely, in the remainder of the paper we will use the following assumptions on given matrices $A \in \mathbb{C}^{n,n}$ and $\Gamma = \Gamma^* \in \mathbb{C}^{m,m}$:

ASSUMPTION 1. *For domains Λ_f^+, Λ_f^- and Λ_f^0 defined in (3.2)–(3.4), $\Lambda(A) \subseteq \Lambda_f^+$ and $\partial\Lambda_f^+ = \partial\Lambda_f^- = \Lambda_f^0$.*

ASSUMPTION 2. *The distance to delocalization $\delta_f^-(A)$ is achieved at a simple singular value of $A - zI$, for z being a solution of (3.9).*

Let us define $s(x,y) := \sigma_n(A-(x+\imath y)I)$. Our aim now is to determine $(\widehat{x},\widehat{y}) \in \mathbb{R}^2$ such that

$$s(\widehat{x},\widehat{y}) = \min\{s(x,y)\ :\ f(x,y) = 0,\ x,y \in \mathbb{R}\}, \tag{4.1}$$

where in the neighborhood of $(\widehat{x},\widehat{y})$ the following equalities hold, see [2, Lemma 3]

$$\begin{array}{rcl} s_x(x,y) &=& -\mathrm{Re}(u^*v),\\ s_y(x,y) &=& \mathrm{Im}(u^*v),\\ s_{xx}(x,y) &=& \sigma u^*Eu + \sigma v^*Fv + 2\mathrm{Re}(v^*(A-zI)Eu) + \sigma^{-1}(\mathrm{Im}(u^*v))^2,\\ s_{xy}(x,y) &=& 2\mathrm{Im}(v^*(A-zI)Eu) + \sigma^{-1}\mathrm{Re}(u^*v)\mathrm{Im}(u^*v),\\ s_{yy}(x,y) &=& \sigma u^*Eu + \sigma v^*Fv - 2\mathrm{Re}(v^*(A-zI)Eu) + \sigma^{-1}(\mathrm{Re}(u^*v))^2. \end{array} \tag{4.2}$$

Here,

$$E = (\sigma^2 I - (A-zI)^*(A-zI))^\dagger \quad \text{and} \quad F = (\sigma^2 I - (A-zI)(A-zI)^*)^\dagger, \tag{4.3}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse [12, Chapter 5], and (σ,u,v) is the minimal singular triplet of $A - zI$ for $z = x + \imath y$.

Thus, we can introduce the Lagrange function $\Phi(x,y,\mu) := s(x,y) + \mu f(x,y)$, where μ is a Lagrange multiplier, and solve (4.1) using Newton's method, which is given by

$$\xi^{(k+1)} = \xi^{(k)} - \left[\nabla^2\Phi\left(\xi^{(k)}\right)\right]^{-1}\nabla\Phi\left(\xi^{(k)}\right), \quad k = 0,1,2,\ldots, \tag{4.4}$$

where $\xi = [x, y, \mu]^T$ and

$$\nabla\Phi = \begin{bmatrix} s_x + \mu f_x \\ s_y + \mu f_y \\ f \end{bmatrix}, \qquad \nabla^2\Phi = \begin{bmatrix} s_{xx} + \mu f_{xx} & s_{xy} + \mu f_{xy} & f_x \\ s_{xy} + \mu f_{xy} & s_{yy} + \mu f_{yy} & f_y \\ f_x & f_y & 0 \end{bmatrix}. \tag{4.5}$$

For the sake of brevity, here and later on, we omit the arguments (x, y, ε). Now, differentiating (3.6) yields the equations

$$\begin{array}{rcl} f_x(x,y) & = & \varphi^T(x,y)\,[D^*\Gamma + \Gamma D]\,\varphi(x,-y), \\ f_y(x,y) & = & \imath\,\varphi^T(x,y)\,[D^*\Gamma - \Gamma D]\,\varphi(x,-y), \\ f_{xx}(x,y) & = & \varphi^T(x,y)\left[D^{2*}\Gamma + 2D^*\Gamma D + \Gamma D^2\right]\varphi(x,-y), \\ f_{xy}(x,y) & = & \imath\,\varphi^T(x,y)\left[D^{2*}\Gamma - \Gamma D^2\right]\varphi(x,-y), \\ f_{yy}(x,y) & = & -\varphi^T(x,y)\left[D^{2*}\Gamma - 2D^*\Gamma D + \Gamma D^2\right]\varphi(x,-y), \end{array} \tag{4.6}$$

where $\varphi(x,y) = \left[1,\, x+\imath y,\, (x+\imath y)^2,\, \ldots,\, (x+\imath y)^{m-1}\right]^T$, and $D = [d_{ij}] \in \mathbb{R}^{m,m}$ is defined such that $d_{ij} = i$ for $2 \le i = j-1 \le m$, and $d_{ij} = 0$ otherwise.

As a consequence, calculating values of $f(x,y)$ and values of its partial derivatives in each iteration step reduces to matrix vector multiplications with a matrix of the size m-by-m. Since for most of the interesting domains it holds $m \le 4$, the computational cost arising from checking the domain constraint is very small compared to the cost of evaluating the function $s(x,y)$ and its derivatives.

We will refer to this primary method for computing the distance to delocalization as Algorithm **eD2D** (**e**xplicit **D**instance **to D**elocalization). This algorithm converges locally quadratically provided that the Hessian $\nabla^2\Phi$ is nonsingular in the limit point, which obviously requires a necessary condition $\nabla f(\widehat{x}, \widehat{y}) \neq 0$.

Algorithm 1 : eD2D

Input: A, Γ, x_0, y_0, μ_0, *tol*

Set $x := x_0$, $y := y_0$, $\mu := \mu_0$
while $\|\triangle\xi\|_2 \ge tol$ **do**
  Compute the singular triplet (σ, u, v) of $A - (x + \imath y)I$;
  Compute E, F from (4.3);
  Compute s_x, s_y, s_{xx}, s_{xy}, s_{yy} using (4.2);
  Compute f, f_x, f_y, f_{xx}, f_{xy}, f_{yy} using (4.6);
  Build gradient $\nabla\Phi$ and Hessian $\nabla^2\Phi$ matrices using (4.5);
  Compute $\triangle\xi$ by solving the linear system $\nabla^2\Psi\,\triangle\xi = -\nabla\Psi$;
  Update $x := x + \triangle\xi_1$, $\quad y := y + \triangle\xi_2$, $\quad \mu := \mu + \triangle\xi_3$;
end while

Output: ε, $x + \imath y$

Solving (3.9) by calculating the minimal singular triplets (ε, u, v) through the singular value decomposition (SVD) is not the best approach when dealing with large (and sparse) matrices arising from practical applications. Therefore, we proceed by extending the recent work of [31], and [1, 10] on the implicit determinant method to replace intensive SVD computations by the LU decompositions which significantly decreases the overall computational cost.

To that end, note that (3.9) can be reformulated as

$$\min\ \varepsilon > 0 \quad \text{s.t.} \quad \begin{array}{l} (A - (x+\imath y)I)v = \varepsilon u, \\ (A^* - (x-\imath y)I)u = \varepsilon v, \\ f(x,y) = 0, \end{array} \quad u, v \in \mathbb{C}^n,\ x, y, \varepsilon \in \mathbb{R}. \tag{4.7}$$

The implicit determinant method essentially bases on the fact that for a given matrix $A \in \mathbb{C}^{n,n}$ and a point $z = x + \imath y \in \mathbb{C}$, $|\varepsilon|$ is a singular value of $A - zI$ if and only if $h(x, y, \varepsilon) = 0$, where $h(x, y, \varepsilon)$ is the last component of a unique solution (when it exists) of the linear system

$$\underbrace{\begin{bmatrix} -\varepsilon I & A - (x + \imath y)I & c_1 \\ A^* - (x - \imath y)I & -\varepsilon I & c_2 \\ c_1^* & c_2^* & 0 \end{bmatrix}}_{M := M(x,y,\varepsilon)} \begin{bmatrix} u(x, y, \varepsilon) \\ v(x, y, \varepsilon) \\ h(x, y, \varepsilon) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4.8}$$

Moreover, see [1, Theorem 2.2], if $\widehat{\varepsilon}$ is a simple singular value of $A-(\widehat{x}+\imath\widehat{y})I$ with corresponding left and right singular vectors $\widehat{u}$, $\widehat{v}$, and $c_1, c_2 \in \mathbb{C}^n$ are chosen such that $c_1^*\widehat{u} + c_2^*\widehat{v} \neq 0$, then the matrix $M(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$ is nonsingular. Hence, (4.8) defines smooth functions $u(x, y, \varepsilon)$, $v(x, y, \varepsilon)$ and $h(x, y, \varepsilon)$ in the neighborhood of $(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$.

We can now reformulate (4.7) locally as follows:

THEOREM 4.1. *Let $A \in \mathbb{C}^{n,n}$ and $\Gamma = \Gamma^* \in \mathbb{C}^{m,m}$. For $\widehat{u}$, $\widehat{v}$ being the left and the right singular vectors corresponding to the singular value $\delta_f^-(A)$ we define $\mathcal{E}$ as a neighborhood of $\delta_f^-(A)$, and we introduce fixed vectors $c_1, c_2 \in \mathbb{C}^n$ such that $c_1^*\widehat{u} + c_2^*\widehat{v} \neq 0$. Then, under the Assumptions 1 and 2, for $(\widetilde{x}, \widetilde{y}, \widetilde{\varepsilon})$ being the solution of the constrained optimization problem:*

$$\min\ \varepsilon^2 \quad s.t. \quad \begin{matrix} h(x, y, \varepsilon) = 0, \\ f(x, y) = 0, \end{matrix} \qquad x, y \in \mathbb{R},\ \varepsilon \in \mathcal{E}, \tag{4.9}$$

the distance to delocalization $\delta_f^-(A)$ of $\Lambda(A)$ from the domain Λ_f^+ is given by $\widetilde{\varepsilon}$, i.e., $\delta_f^-(A) = \widetilde{\varepsilon}$.

Proof. First, let us observe that, under the Assumption 1, (3.9) holds, i.e., $\delta_f^-(A)$ is the smallest minimal singular value of $A - zI$, where $z \in \Lambda_f^0$. Let $\widehat{z} = \widehat{x} + \imath\widehat{y}$ be a complex number for which the minimum of (3.9) is attained. Then

$$f(\widehat{x}, \widehat{y}) = 0, \quad (A - \widehat{z}I)\widehat{v} = \delta_f^-(A)\widehat{u}, \quad \text{and} \quad (A - \widehat{z}I)^*\widehat{u} = \delta_f^-(A)\widehat{v}. \tag{4.10}$$

Since $c_1^*\widehat{u}+c_2^*\widehat{v} \neq 0$, Assumption 2 implies that $M(\widehat{x}, \widehat{y}, \delta_f^-(A))$ is nonsingular, and consequently (4.8) defines a smooth function $h(x, y, \varepsilon)$ in a neighborhood of $(\widehat{x}, \widehat{y}, \delta_f^-(A))$ with $h(\widehat{x}, \widehat{y}, \delta_f^-(A)) = 0$. Therefore, the problem (4.9) is well defined and for its solution $(\widetilde{x}, \widetilde{y}, \widetilde{\varepsilon})$ we have $h(\widetilde{x}, \widetilde{y}, \widetilde{\varepsilon}) = 0$ and $f(\widetilde{x}, \widetilde{y}) = 0$. However, the nonsingularity of $M(\widetilde{x}, \widetilde{y}, \widetilde{\varepsilon})$ implies that $\widetilde{\varepsilon}$ is a singular value of $A - (\widetilde{x} + \imath\widetilde{y})I$, where $\widetilde{x} + \imath\widetilde{y} \in \Lambda_f^0$. This however, according to (3.9), implies $\delta_f^-(A) \leq \widetilde{\varepsilon}$. Finally, since $\widetilde{\varepsilon}$ is the minimum attained in (4.9), we conclude that $\delta_f^-(A) = \widetilde{\varepsilon}$ which completes the proof. □

Theorem 4.1 guarantees that solving (4.9) will provide us with the desired value of distance to delocalization. Hence, as before, we define the Lagrange function of (4.9) by

$$\Psi(x, y, \varepsilon, \zeta, \mu) := \varepsilon^2 + \zeta h(x, y, \varepsilon) + \mu f(x, y), \tag{4.11}$$

where, ζ and μ are Lagrange multipliers. We construct Newton's method for computing the stationary points of (4.9)

$$\xi^{(k+1)} = \xi^{(k)} - \left[\nabla^2\Psi\left(\xi^{(k)}\right)\right]^{-1} \nabla\Psi\left(\xi^{(k)}\right), \quad k = 0, 1, 2, \ldots, \tag{4.12}$$

where $\xi = [x, y, \varepsilon, \zeta, \mu]^T$ and

$$\nabla\Psi = \begin{bmatrix} \zeta h_x + \mu f_x \\ \zeta h_y + \mu f_y \\ 2\varepsilon + \zeta h_\varepsilon \\ h \\ f \end{bmatrix}, \quad \nabla^2\Psi = \begin{bmatrix} \zeta h_{xx} + \mu f_{xx} & \zeta h_{xy} + \mu f_{xy} & \zeta h_{x\varepsilon} & h_x & f_x \\ \zeta h_{xy} + \mu f_{xy} & \zeta h_{yy} + \mu f_{yy} & \zeta h_{y\varepsilon} & h_y & f_y \\ \zeta h_{x\varepsilon} & \zeta h_{y\varepsilon} & 2 + \zeta h_{\varepsilon\varepsilon} & h_\varepsilon & 0 \\ h_x & h_y & h_\varepsilon & 0 & 0 \\ f_x & f_y & 0 & 0 & 0 \end{bmatrix}. \tag{4.13}$$

Expressions for derivatives of f are given in (4.6), whereas formulas for derivatives of h can be obtained by differentiating (4.8):

$$M(x, y, \varepsilon) \begin{bmatrix} u_x & u_y & u_\varepsilon \\ v_x & v_y & v_\varepsilon \\ h_x & h_y & h_\varepsilon \end{bmatrix} = \begin{bmatrix} v & \imath v & u \\ u & -\imath u & v \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.14}$$

and

$$M(x, y, \varepsilon) = \begin{bmatrix} u_{xx} & u_{xy} & u_{x\varepsilon} & u_{yy} & u_{y\varepsilon} & u_{\varepsilon\varepsilon} \\ v_{xx} & v_{xy} & v_{x\varepsilon} & v_{yy} & v_{y\varepsilon} & v_{\varepsilon\varepsilon} \\ h_{xx} & h_{xy} & h_{x\varepsilon} & h_{yy} & h_{y\varepsilon} & h_{\varepsilon\varepsilon} \end{bmatrix} = \tag{4.15}$$

$$\begin{bmatrix} 2v_x & v_y + \imath v_x & u_x + v_\varepsilon & 2\imath v_y & u_y + \imath v_\varepsilon & 2u_\varepsilon \\ 2u_x & u_y - \imath u_x & v_x + u_\varepsilon & -2\imath u_y & v_y - \imath u_\varepsilon & 2v_\varepsilon \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Again, calculating the first and the second order partial derivatives of $h(x, y, \varepsilon)$ reduces to solving linear systems with the matrix $M(x, y, \varepsilon)$. Therefore, we are able to implement (4.12) in Algorithm **iD2D$_0$** (**i**mplicit **D**istance **to D**elocalization) such that only one LU factorization of a Hermitian matrix per Newton step is needed. So far, we have replaced one SVD and two $n \times n$ Moore-Penrose pseudoinverses by one $(2n+1)\times(2n+1)$ *LU* factorization and six forward/backward substitutions. Obviously, this change has significantly reduced the computational cost, but, unfortunately, it does suffer from several drawbacks. Before we go into more details, let us provide the conditions for the convergence of Algorithm iD2D$_0$.

THEOREM 4.2. *Let $A \in \mathbb{C}^{n,n}$, $\Gamma = \Gamma^* \in \mathbb{C}^{m,m}$ satisfy the Assumptions 1 and 2, and let $(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$ be the solution of (4.9). Then, Algorithm iD2D$_0$ converges quadratically to $(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$ for all starting values $(x_0, y_0, \varepsilon_0)$ sufficiently close to $(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$, provided that the following conditions hold:*

(1) $\nabla^2 f(\widehat{x}, \widehat{y}) \neq 0$,

(2) $c_1, c_2 \in \mathbb{C}^n$ are chosen such that $c_1^* \widehat{u} + c_2^* \widehat{v} \neq 0$,

(3) ζ_0, μ_0 are sufficiently close to $-2\,(\|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2)^{-1}\widehat{\varepsilon}$ and $2\,\widehat{\alpha}\,\widehat{\varepsilon}$, respectively,

(4) and $\nabla^2 h(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) - (\|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2) \begin{bmatrix} \widehat{\alpha}\,\nabla^2 f(\widehat{x}, \widehat{y}) & 0 \\ 0 & \widehat{\varepsilon}^{-1} \end{bmatrix}$ is nonsingular,

with $\widehat{u} := u(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$, $\widehat{v} := v(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$, and

$$\widehat{\alpha} := \begin{cases} \dfrac{2Re(\widehat{u}^*\widehat{v})}{(\|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2) f_x(\widehat{x}, \widehat{y})}, & f_x(\widehat{x}, \widehat{y}) \neq 0, \\ -\dfrac{2Im(\widehat{u}^*\widehat{v})}{(\|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2) f_y(\widehat{x}, \widehat{y})}, & f_x(\widehat{x}, \widehat{y}) = 0. \end{cases} \tag{4.16}$$

Proof. Since Algorithm D2D$_0$ is the Newton-type method given in (4.4), it calculates a vector $\widehat{\xi} = [\widehat{x}, \widehat{y}, \widehat{\varepsilon}, \widehat{\zeta}, \widehat{\mu}]^T$ such that $\nabla\Psi(\widehat{\xi}) = 0$, and its local quadratic

Algorithm 2 : iD2D$_0$

Input: A, Γ, c_1, c_2, x_0, y_0, ε_0, ζ_0, μ_0, tol

Set $x := x_0$, $y := y_0$, $\varepsilon := \varepsilon_0$, $\zeta = \zeta_0$, $\mu := \mu_0$;
Set $\triangle\xi = 2tol$;
while $\|\triangle\xi\|_2 \geq tol$ **do**
  Build $M = M(x, y, \varepsilon)$ as in (4.8);
  Determine the LU decomposition $M = LU$;
  Solve the linear system (4.8) to obtain h, u, v;
  Solve the linear system (4.14) to obtain h_x, h_y, h_ε;
  Solve the linear system (4.15) to obtain h_{xx}, h_{xy}, $h_{x\varepsilon}$, h_{yy}, $h_{y\varepsilon}$, $h_{\varepsilon\varepsilon}$;
  Calculate basis φ;
  Compute f, f_x, f_y, f_{xx}, f_{xy}, f_{yy} using (4.6);
  Build gradient $\nabla\Psi$ and Hessian $\nabla^2\Psi$ matrices using (4.13);
  Compute $\triangle\xi$ by solving the linear system $\nabla^2\Psi\,\triangle\xi = -\nabla\Psi$;
  Update $x := x + \triangle\xi_1$, $y := y + \triangle\xi_2$, $\varepsilon := \varepsilon + \triangle\xi_3$, $\zeta := \zeta + \triangle\xi_4$, $\mu := \mu + \triangle\xi_5$;
end while

Output: ε, $x + \imath y$

convergence follows directly from the nonsingularity of the Hessian in the limit vector $\widehat{\xi}$, i.e, $\nabla^2\Psi(\widehat{\xi})$. According to Theorem 4.1, we need to guarantee that the starting vector ξ_0 in (4.4) is sufficiently close to $\widehat{\xi}$ and prove that the 5×5 Hermitian matrix $\nabla^2\Psi(\widehat{\xi})$ is nonsingular.

First, we observe that, due to condition *(1)*, $\widehat{\alpha}$ is well defined. Second, since $h(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = 0$, we have that $c_1^*\widehat{u} + c_2^*\widehat{v} = 1$ and $[\widehat{u}^*,\, \widehat{v}^*,\, 0]\, M(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = [0,\, 0,\, 1]^T$, see (4.8). By multiplying (4.14) with $[\widehat{u}^*,\, \widehat{v}^*,\, 0]$ we get $h_x(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = 2\mathrm{Re}(\widehat{u}^*\widehat{v})$, $h_y(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = -2\mathrm{Im}(\widehat{u}^*\widehat{v})$ and $h_\varepsilon(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = \|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2 > 0$. Now, exploiting the fact that $\nabla\Psi(\widehat{\xi}) = 0$, we obtain

$$\widehat{\zeta} = -\frac{2}{\|\widehat{u}\|_2^2 + \|\widehat{v}\|_2^2}\,\widehat{\varepsilon}, \quad \widehat{\mu} = 2\,\widehat{\alpha}\,\widehat{\varepsilon}, \quad h(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) = 0, \quad \text{and} \quad f(\widehat{x}, \widehat{y}) = 0, \tag{4.17}$$

which guaranties that ξ_0 is sufficiently close to $\widehat{\xi}$.

To prove the nonsingularity of the Hessian matrix, we notice that it can be written as

$$\nabla^2\Psi(\widehat{\xi}) = \begin{bmatrix} B & P \\ P^T & 0 \end{bmatrix},$$

where

$$B = \widehat{\zeta}\,\nabla^2 h(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) + \begin{bmatrix} \widehat{\mu}\,\nabla^2 f(\widehat{x}, \widehat{y}) & 0 \\ 0 & 2 \end{bmatrix} \quad \text{and} \quad P = \left[\begin{array}{c|c} \nabla h(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) & \nabla f(\widehat{x}, \widehat{y}) \\ & 0 \end{array}\right].$$

Condition *(4)* implies that B is nonsingular, while $\mathrm{rank}(P) = 2$ due to $h_\varepsilon(\widehat{x}, \widehat{y}, \widehat{\varepsilon}) > 0$. Finally, the equality $\det[\nabla^2\Psi(\widehat{\xi})] = \det[B]\det[P^T B^{-1} P]$ implies the nonsingularity of $\nabla^2\Psi(\widehat{\xi})$, which completes the proof. □

According to Theorem 4.2, Algorithm iD2D$_0$ exhibits a local quadratic convergence for an appropriately chosen starting point, while the computed value of ε is only a stationary point of (4.11). Both algorithms eD2D and iD2D$_0$, as Newton-type

methods, may not produce the distance to delocalization, but, instead, return a larger value of ε, such that $\Lambda_\varepsilon(A) \cap \Lambda_f^- \neq \emptyset$. These suboptimal results may arise due to the general relationship between geometries of the pseudospectrum and the domain. This makes the choice of a good starting point $x_0 + \imath y_0$ hard, even if the eigenvalues of A are known. This phenomenon will be illustrated in the following section. However, unlike Algorithm eD2D, Algorithm iD2D$_0$ can also produce suboptimal solutions since $h(x, y, \varepsilon) = 0$ implies that $|\varepsilon|$ is a singular value, but not necessary the minimal one. To overcome this drawback, we develop an implicit version of the D2D algorithm.

Let us now take a point $z_0 = x_0 + \imath y_0$ in the complex plane which is sufficiently close to the solution $\widehat{z} = \widehat{x} + \imath\widehat{y}$ of (3.9). In order to run Algorithm iD2D$_0$, we will still need to specify $c_1, c_2 \in \mathbb{C}^n$ and good starting values for $\varepsilon_0, \zeta_0, \mu_0 \in \mathbb{R}$. Since, under Assumption 2, see [2], the singular value of $A - zI$ and its corresponding left and right singular vectors are differentiable functions of z in the neighborhood of $\widehat{z}$, Theorem 4.2 provides good starting values. Namely, we can compute the minimal singular triplet $(\varepsilon_0, u_0, v_0)$ of $A - (x_0 + \imath y_0)I$ and set $c_1 = u_0/2$ and $c_2 = v_0/2$. Then, $0 \neq c_1^*\widehat{u} + c_2^*\widehat{v} \approx c_1^*u_0 + c_2^*v_0 = 1$ and condition (*1*) of Theorem 4.2 is fulfilled. Similarly, condition (*2*) gives appropriate starting values for the Lagrange multipliers $\zeta_0 = -\widehat{\varepsilon}$ and $\mu_0 := 2\mathrm{Re}(\widehat{u}_0^*\widehat{v}_0)/f_x(x_0, y_0)$ for $f_x(x_0, y_0) \neq 0$, and $\mu_0 := -2\mathrm{Im}(u_0^*v_0)/f_y(x_0, y_0)$ otherwise. With such constructed starting values, given matrices A and Γ, and computed tolerance *tol*, we can run Algorithm iD2D$_0$ by specifying an initial point in the complex plane. As a result, we get $\widehat{z} = \widehat{x} + \imath\widehat{y}$ and $\widehat{\varepsilon}$, which is a singular value of $A - \widehat{z}I$. Unfortunately, we cannot guarantee that $\widehat{\varepsilon}$ is the minimal singular value of $A - \widehat{z}I$. Namely, taking into account that $\partial\Lambda_{\widehat{\varepsilon}}(A)$ consists of the outermost closed curves of $\{(x, y) \in \mathbb{R}^2 : h(x, y, \widehat{\varepsilon}) = 0\}$, it may happen that a chosen starting point was closer to the place where some of the inner curves touch the boundary of the domain Λ_f^+. As a consequence, we may end up in a suboptimal value of $\widehat{\varepsilon} > \delta_f^-(A)$. A remedy for this problem is to compare $\widehat{\varepsilon}$ with $\sigma_n(A - \widehat{z}I)$, and to restart the algorithm if necessary. Using these two tricks, we have the following improved implicit D2D Algorithm, denoted by **iD2D**.

Algorithm 3 : iD2D

Input: A, Γ, x_0, y_0, *tol*

Compute the minimal singular triplet $(\varepsilon_0, u_0, v_0)$ of $A - (x_0 + \imath y_0)I$;
Set $\varepsilon := \varepsilon_0 + 2tol$;
while $|\varepsilon - \varepsilon_0| \geq tol$ **do**
  Set $c_1 = u_0/\sqrt{2}$,$c_2 = v_0/\sqrt{2}$ and $\zeta_0 = -\varepsilon_0$;
  Compute $f_x(x_0, y_0)$ and $f_y(x_0, y_0)$ using (4.6);
  if $f_x(x_0, y_0) > tol$ **then**
    Set $\mu_0 = 2\varepsilon_0\,\mathrm{Re}(u_0^*v_0)/f_x(x_0, y_0)$;
  else
    Set $\mu_0 = -2\varepsilon_0\,\mathrm{Im}(u_0^*v_0)/f_y(x_0, y_0)$;
  end if
  Run the Algorithm iD2D$_0$(A, Γ, c_1, c_2, x_0, y_0, ε_0, ζ_0, μ_0) to obtain ε, x and y;
  Set $x_0 := x$, $y_0 := y$;
  Compute the minimal singular triplet $(\varepsilon_0, u_0, v_0)$ of $A - (x_0 + \imath y_0)I$;
end while

Output: ε, $x + \imath y$

Although the computation of non-minimal singular values in iD2D$_0$ can be fixed by restarts implemented in iD2D, another way to solve this issue is to stabilize the Newton iterations by step size control. The standard, state of the art, stabilization techniques use back tracking with the Armijo rule [3] and the Wolfe conditions [35, 36]. However, these techniques require the recomputation of the Lagrange function (4.11) and its gradient for every contraction step, and result in an increased number of LU factorizations and forward/backward substitutions. Fortunately, since computing the values of f is extremely cheap in comparison to the computation of h, step size control based on values of f seems to be a much better choice. In fact, we are iteratively approaching the point in the complex plane which corresponds to the coalescence of the domain boundary Λ_f^0 and the outermost curves of $\{(x,y) \in \mathbb{R}^2 : h(x,y,\widehat{\varepsilon}) = 0\}$. Therefore, if our step size control assures that we move in the steepest descent direction while staying out of the domain Λ_f^+, we automatically enforce computations of ε values that correspond to the minimal singular values. This leads to the final Algorithm **iD2D$_\mathbf{d}$** (**i**mplicit **D**istance **to** **D**elocalization - **d**amped). To simplify the presentation we state only the part of the step size control using step contraction parameter $\beta \in (0,1)$ and the smallest step parameter $\tau \in (0,1)$ which replaces line 13 in the Algorithm iD2D$_0$:

Lines 13a-13f for iD2D$_0$ (damping of the descent step size):

Set $\gamma = 1$;
while $f < 0$ and $\tau < \gamma$ **do**
  Update $\gamma = \gamma\beta$;
  Compute f using (3.6);
end while
Update $x := x+\gamma\triangle\xi_1$, $y := y+\gamma\triangle\xi_2$, $\varepsilon := \varepsilon+\gamma\triangle\xi_3$, $\zeta := \zeta+\gamma\triangle\xi_4$, $\mu := \mu+\gamma\triangle\xi_5$;

In the following section, we will illustrate our algorithms for computing the distance to delocalization using several test examples. In the case when it coincides with the distance to instability, we will compare our results with the algorithm of [10], denoted as FS, since they are both based on Newton's method. In addition, we will discuss some strategies for choosing the proper starting points, which are crucial for the convergence of the method. It is important to note that the computed values may not coincide with the distance to delocalization and that, contrary to the case of the left half-pane, up to now there is no reasonable method to test if the solution is a global minimum. The distance to instability test in [17], based on computing imaginary eigenvalues of Hamiltonian matrices, was used in almost all the state of the art algorithms. Unfortunately, this approach cannot easily be generalized to the more complicated domains introduced in this paper. Still missing is a simple and efficient test for checking whether the ε-pseudospectrum crosses an algebraic curve of the form Λ_f^0.

5. Numerical examples. In this section, we start with three example problems from [10] for computing the distance to instability. Since that algorithm is also based on Newton's method and the implicit determinant approach, it seems reasonable to compare it with our algorithms with a special choice of the domain, namely, the open left half-plane. More precisely, we will present a comparison of the implicit D2D algorithm (iD2D) and the FS algorithm, and show that iD2D can be considered as a suitable generalization of the FS method to domains different than the open left half-plane. To ensure the global optimum in the outer iteration of iD2D, we use the

same test as in FS, see [10], which, unfortunately, is restricted to the case of the open left half-plane. For a detailed discussion, see [10].

All algorithms were implemented in MATLAB version 8.1 (R2013a) running on an Intel® Core™ 2 DUO CPU E6850 and tested with some matrices from the MATRIX MARKET repository [5] and EIGTOOL [37].

EXAMPLE 1. *Let A be the* `bwm200.mtx` *matrix from the* MATRIX MARKET *repository. The size of the matrix is $n = 200$ and it originates from a Brusselator wave model in a chemical reaction. The starting point for both FS and iD2D algorithm is chosen according to [10] as $z_0 = 2.139497522076343\mathrm{i}$, and the tolerance is set to $tol = 10^{-10}$.*

In Table 5.1 we present the convergence behavior of both algorithms. After initialization of iD2D, which consists of computing the minimal singular triplet of $A - z_0 I$ and setting up starting values ε_0, ζ_0 and μ_0, both algorithms were executed with the same starting values. We present only the last 8 digits of $y^{(i)}$ ($y^{(0)} = 2.1394975\mathbf{22076343}$) and 3 digits of $\varepsilon^{(i)}$ ($\varepsilon^{(0)} = 8.240972\mathbf{726}e-06$ for algorithm iD2D), which are the only ones varying during the iterations. The values of $x^{(i)}$ are omitted since they are equal to zero (up to machine precision).

i	$y^{(i)} = 2.1394975\ldots$		$\varepsilon^{(i)} = 8.240972\ldots$		Error$^{(i)}$	
	FS	iD2D	FS	iD2D	FS	iD2D
0	22076343	22076343	0	726	-	-
1	22045502	21953098	700	726	4.19e-06	8.24e-06
2	22014727	22014689	687	697	3.83e-07	6.16e-11
3	22014739	-	689	-	1.63e-10	-
4	22014746	-	691	-	8.10e-11	-

Table 5.1: Results for Brusselator wave model - Example 1. Algorithm FS is compared to algorithm iD2D.

As we can see, algorithm iD2D converges in two iterations, whereas FS needed four iterations to obtain the same distance to instability $\widehat{\varepsilon} = 8.240972e-06$. This demonstrates the advantage of formulating the optimization problem of distance to instability using the Hermitian function f over the formulation via Hamiltonian eigenvalue problems as it was done in [10]. Also, the CPU times of FS ($1.29s$) and iD2D ($0.7s$) indicate a better performance of iD2D.

EXAMPLE 2. *Our second example is the Tolosa matrix* `tols340.mtx` *of size $n = 340$ from the* MATRIX MARKET *repository. It is a highly nonnormal matrix used in the stability analysis of a flying airplane. Again, the starting point $z_0 = 155.9999219999809\imath$ and tolerance $tol = 10^{-12}$ are chosen as in [10]. Other starting values are obtained through initialization of iD2D.*

Table 5.2 presents the last 10 digits of $y^{(i)}$ and 7 digits of $\varepsilon^{(i)}$, where $y^{(0)} = 155.999\mathbf{9219999809}$ and $\varepsilon^{(0)} = 0.0019997\mathbf{9129104}$ for algorithm iD2D. Here, both algorithms converge in four steps to the same $\widehat{\varepsilon}$, however, the CPU times FS ($3.7s$) and iD2D ($2.85s$) again indicate better performance of iD2D.

EXAMPLE 3. *Both algorithms are tested with matrix* `rdb450.mtx` *of dimension 450 from the* MATRIX MARKET *repository, which originates from a reaction-diffusion Brusselator model. Starting point $z_0 = 1.610747974050403\imath$ and tolerance $tol = 10^{-12}$ are used. Analogously to the previous example, this gives the same values*

i	$y^{(i)} = 155.999\ldots$		$\varepsilon^{(i)} = 0.00199979\ldots$		$\text{Error}^{(i)}$	
	FS	iD2D	FS	iD2D	FS	iD2D
0	9219999809	9219999809	0	7129104	-	-
1	8829972845	7659892313	6887893	7129104	1.00e-03	2.01e-03
2	8439945398	8439945868	6825390	6137865	1.60e-06	7.80e-05
3	8439945282	8439945282	6887893	6637884	3.13e-11	3.58e-10
4	8439945282	8439945282	6887893	6887893	3.79e-16	1.04e-14

Table 5.2: Results for Example 2, Tolosa matrix. Algorithms FS and iD2D are compared.

$\widehat{\varepsilon} = 0.084277384643143$ *and* $\widehat{z} = 1.593892567251319\imath$*, while CPU times, FS* $(8.2s)$ *and iD2D* $(6.81s)$*, show again a slightly better performance of iD2D.*

Based on these and other tests, we observe that our method performs as good as the state of the art methods for computing the distance to instability. But more importantly, the distance to delocalization algorithms presented here are the first approaches designed for determining robustness of other domains of spectral inclusion. In the following examples, we illustrate the behaviour of the new algorithms for four different domains: an annulus given by Γ_a as in (3.8), a cissoid of Diocles given by Γ_d and the area between two hyperbolas given by Γ_h, where

$$(5.1)\ \Gamma_d = \begin{bmatrix} 0 & 0 & a/2 \\ 0 & -a & -1 \\ a/2 & -1 & 0 \end{bmatrix}, \qquad \Gamma_h = -\begin{bmatrix} -4a^2b^2 & 0 & a^2+b^2 \\ 0 & 2(b^2-a^2) & 0 \\ a^2+b^2 & 0 & 0 \end{bmatrix}.$$

Example 4. *In this example we will consider the perturbed motion of a rocket, taking into account the elastic oscillations of its airframe as a straight flexible nonuniform rod. The governing equations of this dynamical system are given in [24, Example 3.1.2.]. The domain of interest for the location of the "stable" dynamical system is chosen to be a cissoid of Diocles* Γ_d *in (5.1) with* $a = 0.1$*. Starting with an unstable sparse matrix* A *of size* $n = 9$*, controller* K *is constructed using Theorem 3.1 and its related theory. Afterwards, a matrix* $M = A + BKC$*, whose eigenvalues are located in the domain bordered by the cissoid of Diocles, is defined. Table 5.3 contains the nonzero elements of the matrices* $A \in \mathbb{R}^{9,9}$*,* $B \in \mathbb{R}^{9,1}$ *and* $C \in \mathbb{R}^{3,9}$*, while* $K = [1.0401, 1.5558, -0.0177]$.

As expected, the choice of a suitable starting point is fundamental to avoid being trapped in a local minimum. This situation is illustrated in Figure 5.1 for two starting points $z_0 = 0.01\imath$ (on the left) and $z_0 = 5\imath$ (on the right). In the latter case, a local minimum is attained, since the pseudospectrum has already crossed the boundary around 0. Indeed, Table 5.4 shows that the computed value $\widehat{\varepsilon}$ for starting point $z_0 = 5\imath$ is much larger than the one computed for $z_0 = 0.01\imath$. However, testing this in practice is still a computationally challenging task, contrary to the case when the domain is an open left-half plane.

For all three algorithms eD2D, iD2D, and iD2D$_d$, finding a good starting point is not an easy task. However, for certain domains one can sample closely outside of the domain Λ_f^+, following the boundary Λ_f^0 by a regularly spaced set of points. Then, one computes the minimal singular value of $A - zI$ for each sample point z, and starts an algorithm in the point corresponding to the smallest minimal singular value. In the

i,j	a_{ij}	i,j	a_{ij}	i,j	a_{ij}	i,j	a_{ij}	i	b_i
1,2	1	3,1	-0.0458	3,8	7e-4	7,6	-169	2	-1.138
2,3	0.2165	3,2	1	4,5	1	7,7	-0.13	3	-0.0348
2,4	-0.0356	3,3	-0.0133	5,4	-29.81	8,9	1	5	29.56
2,6	-0.0299	3,4	4e-4	5,5	-0.0546	9,8	-334.3	7	47.25
2,8	-0.027	3,6	6e-4	6,7	1	9,9	-0.1828	9	16.4

i,j	c_{ij}	i,j	c_{ij}	i,j	c_{ij}	i,j	c_{ij}	i,j	c_{ij}
1,1	1	1,8	-0.124168	2,7	-0.082347	3,4	-0.73673	3,7	0.041957
1,4	0.02462	2,2	1	2,9	-0.08976	3,5	-1.1663e-5	3,8	161.21166
1,6	0.06918	2,5	-0.04819	3,3	4.368	3,6	53.7935	3,9	0.08851

Table 5.3: The nonzero elements of matrices $A \in \mathbb{R}^{9,9}$, $B \in \mathbb{R}^{9,1}$ and $C \in \mathbb{R}^{3,9}$ of Example 4.

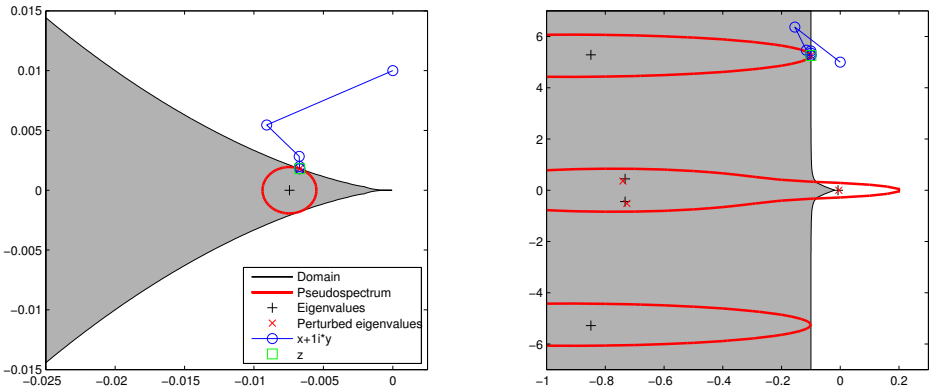

Fig. 5.1: Results of iD2D for Example 4 with starting point $z_0 = 0.01\imath$ (left) and $z_0 = 5\imath$ (right).

following, we have implemented a simple sampling strategy using a built-in MATLAB function `contourc` and computed the corresponding minimal singular triplets using the MATLAB `svd` function. However, if one deals with large and sparse matrix, a better option would be to use the `svds` function based on the implicitly restarted Arnoldi iteration. The same holds for the implementation of the eD2D algorithm for sparse matrices.

EXAMPLE 5. *Let* $A = 0.3 \cdot demmel(10)$, *where* $demmel(10)$ *is the Demmel matrix of dimension* $n = 10$ *from the* EIGTOOL. *We test algorithms eD2D, iD2D and* $iD2D_d$ *(with parameters* $\beta = 0.8$ *and* $\tau = 0.1$*) for the annulus domain* Γ_a *in (3.8) with* $R = 1$ *and* $r = 0.1$.

In the preliminary phase, we sample the region around the domain using ten points per contour and compute the corresponding singular triplets (0.66s of CPU time). In Table 5.5 we present the results of our algorithms. The last two columns of the table contain the condition numbers of the system matrices $M(\widehat{x}, \widehat{y}, \widehat{\varepsilon})$ and the Hessian matrices. Although these are relatively high due to the ill-conditioning of the eigenvalues of the Demmel matrix, our methods still converge. Also, since the size

Starting point	Initialization	"Inner" iterations		"Outer" iterations		Total	$\widehat{\varepsilon}$
	Time (s)	Quantity	Time (s)	Quantity	Time (s)	Time (s)	
$0.01\imath$	6.98e-4	8	0.0097	1	4.11e-4	0.0108	9.3e-4
$5\imath$	2.2e-4	9	0.0084	1	3.97e-4	0.0090	0.13

Table 5.4: Number of iterations, CPU times, and $\widehat{\varepsilon}$ for Example 4 tested with iD2D and two different starting points.

of the problem is small ($n = 10$), the CPU times for the explicit and implicit D2D algorithms are almost the same, which makes eD2D a better choice in this example. The behavior of all the three algorithms is illustrated in Figure 5.2. The sampling points are marked as solid dots, and the other objects are labeled according to Figure 5.1: the domain Λ_f^+ is shaded, the boundary Λ_f^0 is plotted with a solid line, the resulting pseudospectrum $\Lambda_{\widehat{\varepsilon}}(A)$ with a thick solid line, the spectrum $\Lambda(A)$ is marked with $+$, the spectrum of the resulting perturbed matrix $\Lambda(A - \widehat{\varepsilon}\widehat{u}\widehat{v}^*)$ with $\times$, the contact point $\widehat{z}$ with $\square$, and the points $x + \imath y$ from the consecutive iterations with $\circ$. It is interesting to note here that the point where distance to delocalization is achieved is away of the place where the boundary of the domain is closest to the spectrum of the matrix. Namely, $\widehat{z} = -1$ belongs to the outer circle of the annulus while the eigenvalues of the matrix (all equal to 0.3) are much closer to the inner circle. This serves as n example that the strategy of choosing a starting point that was introduced in [10] is not always a suitable one and that sometimes sampling of the boundary is crucial to the global convergence.

Algorithm	Iterations		CPU time		$\widehat{\varepsilon}$	$\widehat{z}$	$\kappa(M)$	$\kappa(H)$
	Inner	Outer	Inner	Outer				
eD2D	16	-	0.012	-	8.8200e-6	-1	-	2.1e9
iD2D	7	1	4.7e-3	3.0e-4	8.8200e-6	-1	2.0e8	2.6e9
iD2D_d	12	1	8.2e-3	3.2e-4	8.8200e-6	-1	2.0e8	2.6e9

Table 5.5: Results for the Demmel matrix with the annulus domain.

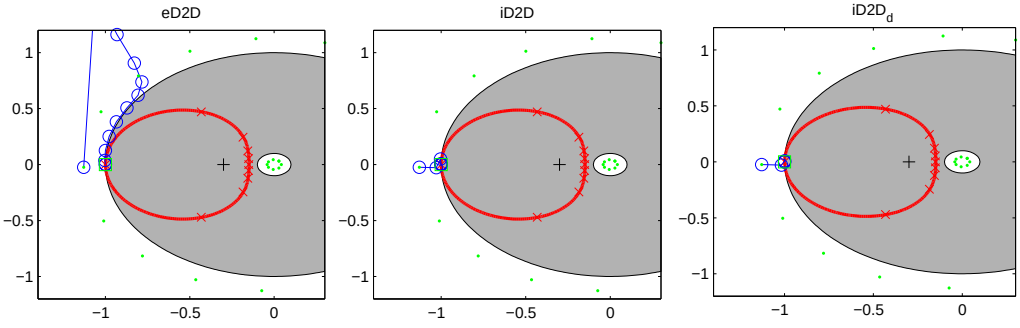

Fig. 5.2: Demmel matrix with annulus domain, $R = 1$ and $r = 1/10$.

EXAMPLE 6. *A particularly challenging example for the distance to delocalization is the "Twisted" matrix A from the* EIGTOOL MATLAB *package [37]. This matrix*

has an exponentially strong degree of nonnormality, with pseudomodes in the form of wave packets. Its pseudospectrum $\Lambda_\varepsilon(A)$ grows the fastest around zero. We will test our algorithms using this matrix and the domain Λ_f^+ defined as Γ_h of (5.1) with $a = b = 1$.

The distance to delocalization $\delta_f^-(A) = 6.35606398911156\text{e-}6$ in this example is attained in $\widehat{z} = 1$ and $\widehat{z} = -1$. Moreover, the second smallest singular value of $A - \widehat{z}I$ is $6.35606401102379\text{e-}6$, which makes the problem almost nongeneric, i.e., the resulting singular value is almost double, since the difference between the two smallest ones is only $2.19e-14$ (smaller than the used tolerance $tol = 10^{-12}$). However, all three algorithms computed the exact value without breaking down. Again, ten sampling points per contour were taken and the corresponding minimal singular values were computed, all in 0.88s of CPU time. Table 5.6 presents detailed results for each iteration step. Approaching the double singular value successfully is reflected in the conditioning of the matrix M. As before, Figure 5.3 illustrates the results.

Algorithm	Iterations		CPU time		$\widehat{\varepsilon}$	$\widehat{z}$	$\kappa(M)$	$\kappa(H)$
	Inner	Outer	Inner	Outer				
eD2D	14	-	0.35	-	6.3561e-6	-1	-	7.5e9
iD2D	35	1	0.42	7.5e-3	6.3561e-6	-1	1.6e14	3.5e12
iD2D$_d$	144	1	1.9	7.4e-3	6.3561e-6	-1	1.7e14	1.3e11

Table 5.6: Results for the Twisted matrix with a hyperbolic domain.

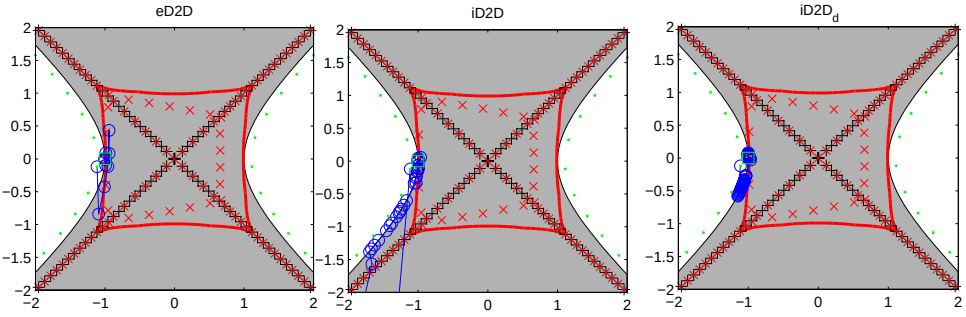

Fig. 5.3: Twisted matrix with a hyperbolic domain.

Finally, we remark that all three algorithms eD2D, iD2D, and iD2D$_d$, can be easily adapted to domains Λ_f^+ which are not constructed using the standard basis. The only changes involve computing a gradient and Hessian of the function f that defines the domain. Omitting the details, we provide a test example that comes from the analysis of the vibration of a wing in the air stream. Namely, for A being a standard linearization of the matrix quadratics of the example *wing* from [4], we compute the distance to delocalization from the nonstandard domain Λ_f^+ (domain with a nonstandard basis φ) defined by $f(x, y) = -x + y^2 - a^2 + \sqrt{x^2 + (y^2 + a^2)^2}$, where $a = 2.2$. Note that this domain contains both the stable modes and the unstable modes with frequencies out of the range given by the parameter a, see the left picture of Figure 5.4. Starting with $z_0 = 1 + 0.2\imath$, while eD2D and iD2D brake down, iD2D$_d$ with damping parameters $\beta = 0.5$ and $\tau = 0.1$, converges in 22 inner and

1 outer iteration taking of $0.014s$ of CPU time producing distance to delocalization $\widehat{\varepsilon} = 0.067738021782556$. This is illustrated in Figure 5.4, where the picture on the right is a zoom of the point where the distance to delocalization is achieved.

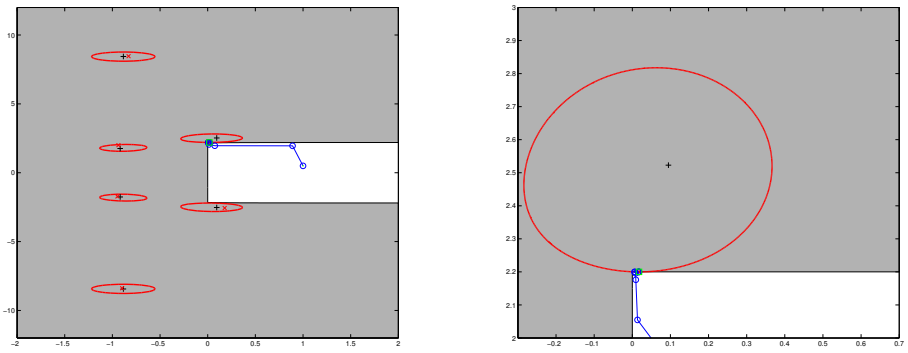

Fig. 5.4: Results of algorithm iD2D with starting point $z_0 = 1$, for the linearization of the quadratics from a wing example, [4].

6. Conclusions and open problems. In this paper we have investigated a class of matrix nearness problems that naturally generalizes the well-established and extensively studied concepts of distance to instability and distance to stability. Based on motivations arising from various applications of the spectral theory of matrices, we have formulated two matrix nearness problems - the distance to delocalization and the distance to localization, and proposed computational methods to determine a solution of the first one.

While the distance to localization was only introduced, the distance to delocalization was studied in more details. In particular, we studied the robustness of the Lyapunov-type spectral inclusions presented in [24] and used the pseudospectral approach to develop algorithms for their effective computations. In the special case of the open left-half plane, when the distance to delocalization coincides with the distance to instability, we have compared the proposed algorithms with the state of the art method from [10]. Moreover, several interesting benchmark problems of practical relevance were used to illustrate their behavior.

For matrices of medium size we have presented the eD2D algorithm that converges to the distance to delocalization, provided that a good starting point in the complex plane is chosen. The boundary sampling technique used in Section 5 was successful in our examples, however, it suffers from a high computational cost for domains with challenging geometries. Therefore, in the general case, the question of choosing a good starting point without expensive computations, remains open.

Another important open problem concerns testing whether eD2D converges to a global or a local solution. One can determine if the ε-pseudospectrum, for a given ε, crosses the boundary of a given domain or not, only in the case of the open left half-plane. In this specific case, the method of [17], based on the computation of Hamiltonian eigenvalues, can be used to check if the obtained solution is a global minimum. While it is simple to extend this approach to any open half-plane or open disk using polar coordinates in the complex plane, other Lyapunov-type domains

remain a challenging task.

Finally, equations (4.2) for the explicit computation of the derivatives of singular values, can be used only under the assumption that the singular value is simple. Although this is generically the case, in some applications this assumption will not be satisfied and different algorithms need to be developed.

Since algorithm eD2D is computationally too demanding for large (and possibly sparse) matrices, we have developed an implicit D2D algorithm based on the implicit determinant method, [1, 10, 31], which avoids SVD, Hermitian or Hamiltonian eigenvalue computations. The use of this implicit approach has increased the sensitivity of the algorithm, therefore, we have provided an efficient step size control that is much cheaper than the commonly used backtracking line search in Newton's methods. Again, choosing a good starting point and assuring that the solution is the global minimum, is a challenging task. Possible improvements of both the explicit and the implicit method may be achieved by replacing the Newton's method for the constrained optimization by a more advanced techniques like tunneling or filtering.

All the open problems discussed now and throughout the paper, together with many interesting applications of the introduced matrix nearness problems, are the subject of further research.

7. Acknowledgements. The authors would like to thank V. Mehrmann for inspiring discussions and pointing out interesting applications of the presented work. V. R. Kostić is partially supported by Ministry of Science, Research Grant 174019, Provincial Secretariat for Science of Vojvodina, Research Grants 1850 and 2492, and the IMU Berlin Einstein Foundation and Berlin Mathematical School Fellowships for 2013/14. A. Międlar is partially supported by the DFG SFB 1029 and J. Stolwijk by the ERC Advanced Grant: "Modeling, Simulation and Control of Multi-Physics Systems".

REFERENCES

[1] R. O. Akinola, M. A. Freitag, and A. Spence. The calculation of the distance to a nearby defective matrix. *Numerical Linear Algebra with Applications*, pages n/a–n/a, 2013.

[2] R. Alam, S. Bora, R. Byers, and M. L. Overton. Characterization and construction of the nearest defective matrix via coalescence of pseudospectral components. *Linear Algebra Appl.*, 435(3):494–513, 2011.

[3] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16:1–3, 1966.

[4] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and c. Tisseur Fran˙Nlevp : A collection of nonlinear eigenvalue problems. Technical report, Manchester Institute for Mathematical Sciences School of Mathematics, 2011.

[5] B. Boisvert, R. Pozo, K. Remington, B. Miller, and R. Lipman. Matrix Market repository. `http://math.nist.gov/MatrixMarket/`.

[6] J. V. Burke, A. S. Lewis, and M. L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA J. Numer. Anal.*, 23(3):359–375, 2003.

[7] R. Byers. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM J. Sci. Statist. Comput.*, 9(5):875–881, 1988.

[8] I. I. Dikin. Iterative solutions of problems of linear and quadratic programming. *Soviet Math. Dok.*, 8:674–675, 1967.

[9] I. Dobson, J. Zhang, S. Greene, H. Engdahl, and P. Sauer. Is strong modal resonance a precursor to power system oscillations? *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48:340–349, 2001.

[10] M. A. Freitag and A. Spence. A Newton-based method for the calculation of the distance to instability. *Linear Algebra Appl.*, 435(12):3189–3205, 2011.

[11] S. Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 6:749–754, 1931.

[12] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[13] N. Guglielmi, D. Kressner, and C. Lubich. *Low Rank Differential Equations for Hamiltonian Matrix Nearness Problems*. Oberwolfach preprints. MFO, 2013.

[14] N. Guglielmi and C. Lubich. Differential equations for roaming pseudospectra: paths to extremal points and boundary tracking. *SIAM J. Numer. Anal.*, 49(3):1194–1209, 2011.

[15] N. Guglielmi and C. Lubich. Erratum/addendum: Differential equations for roaming pseudospectra: paths to extremal points and boundary tracking. *SIAM J. Numer. Anal.*, 50(2):977–981, 2012.

[16] N. Guglielmi and M. L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1166–1192, 2011.

[17] C. He and G. A. Watson. An algorithm for computing the distance to instability. *SIAM J. Matrix Anal. Appl.*, 20(1):101–116 (electronic), 1999.

[18] J. J. Hench, C. He, V. Kučera, and V. Mehrmann. Dampening controllers via a Riccati equation approach. *IEEE Trans. Automat. Control*, 43(9):1280–1284, 1998.

[19] N. J. Higham. Matrix nearness problems and applications. In *Applications of matrix theory (Bradford, 1988)*, volume 22 of *Inst. Math. Appl. Conf. Ser. New Ser.*, pages 1–27. Oxford Univ. Press, New York, 1989.

[20] V. Kostić. On general principles of eigenvalue localisations via diagonal dominance. Technical report, University of Novi Sad, 2014. Submitted.

[21] D. Kressner and B. Vandereycken. Subspace methods for computing the pseudospectral abscissa and the stability radius. Technical report, ANCHP-MATHICSE, Mathematics Section, 'Ecole Polytechnique F'ed'erale de Lausanne, 2012.

[22] P. Lancaster and M. Tismenetsky. *The theory of matrices*. Computer Science and Applied Mathematics. Academic Press, Inc., Orlando, FL, second edition, 1985.

[23] A. M. Lyapunov. *The general problem of the stability of motion*. Taylor & Francis Ltd., London, 1992. Translated from Edouard Davaux's French translation (1907) of the 1892 Russian original and edited by A. T. Fuller.

[24] A. Mazko. *Matrix Equations, Spectral Problems and Stability of Dynamic Systems*. Stability, Oscillations and Optimization of Systems. Cambridge Scientific Publishers, 2008.

[25] V. Mehrmann and C. Schröder. Nonlinear eigenvalue and frequency response problems in industrial practice. *Journal of Mathematics in Industry*, 1(1):7, 2011.

[26] E. Mengi and M. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA J. of Numer. Anal.*, 25(4):648–669, 2005.

[27] F.-X. Orbandexivry, Y. Nesterov, and P. Van Dooren. Nearest stable system using successive convex approximations. *Automatica J. IFAC*, 49(5):1195–1203, 2013.

[28] H. Ouyang. Moving-load dynamic problems: A tutorial (with a brief overview). *Mechanical Systems and Signal Processing*, 25(6):2039 – 2060, 2011. Interdisciplinary Aspects of Vehicle Dynamics.

[29] H. Ouyang, W. Nack, Y. Yuan, and F. Chen. Numerical analysis of automotive disc brake squeal: a review. *Int. J. Vehicle Noise and Vibration*, Volume 1, Nos. 3/4:207–231, 2005.

[30] P. W. Sauer and M. A. Pai. *Power System Dynamics and Stability*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

[31] A. Spence and C. Poulton. Photonic band structure calculations using nonlinear eigenvalue techniques. *J. Comput. Phys.*, 204(1):65–81, 2005.

[32] L. N. Trefethen and M. Embree. *Spectra and pseudospectra*. Princeton University Press, Princeton, NJ, 2005. The behavior of nonnormal matrices and operators.

[33] C. Van Loan. How near is a stable matrix to an unstable matrix? In *Linear algebra and its role in systems theory (Brunswick, Maine, 1984)*, volume 47 of *Contemp. Math.*, pages 465–478. Amer. Math. Soc., Providence, RI, 1985.

[34] R. S. Varga. *Geršgorin and his circles*, volume 36 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2004.

[35] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11:226–235, 1969.

[36] P. Wolfe. Convergence conditions for ascent methods. II. Some corrections. *SIAM Rev.*, 13:185–188, 1971.

[37] T. G. Wright. EigTool. `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`, 2002.