Fast Interpolation-based Globality Certificates for Computing Kreiss Constants and the Distance to Uncontrollability

Tim Mitchell[*]

October 2nd, 2019

Abstract

The Kreiss constant of a matrix and the distance to uncontrollability can both be defined by global minimization problems of certain singular value functions in two real variables, which often have multiple local minima. The state-of-the-art for computing both of these quantities uses optimization to first find minimizers and then computes globality certificates to either assert that a given minimizer is a global one, or when not, provide new starting points for another round of optimization. These existing globality certificates are expensive to compute, which limits them to rather small problems, and for Kreiss constants, they also have high memory requirements. In this paper, we propose alternative globality certificates for both Kreiss constants and the distance to uncontrollability, based on the idea of building interpolant approximations to certain one-variable distance functions. Our new certificates can be orders of magnitude faster to compute, have relatively low memory requirements, and seem to be more reliable in practice.

Notation: $\|\cdot\|$ denotes the spectral norm, $\sigma_{\min}(\cdot)$ the smallest singular value, $\Lambda(\cdot)$ the spectrum, and (A,B) the matrix pencil $A-\lambda B$, with $\Lambda(A,B)$ denoting the spectrum of matrix pencil (A,B).

1 Introduction

The Kreiss Matrix Theorem [Kre62], after being refined by many authors over nearly thirty years, says that for any matrix $A \in \mathbb{C}^{n \times n}$ [TE05, Theorem 18.1]

$$\mathcal{K}(A) \leq \sup_{k \geq 0} \|A^k\| \leq \mathrm{e}n\mathcal{K}(A), \tag{1.1}$$

where the *Kreiss constant* $\mathcal{K}(A)$ has two equivalent definitions [TE05, p. 143]

$$\begin{aligned} \mathcal{K}(A) &= \sup_{z \in \mathbb{C}, |z|>1} (|z|-1)\|(zI-A)^{-1}\| && (1.2a) \\ &= \sup_{\varepsilon > 0} \frac{\rho_\varepsilon(A)-1}{\varepsilon}, && (1.2b) \end{aligned}$$

and the ε-*pseudospectral radius* $\rho_\varepsilon(\cdot)$ is defined by

$$\begin{aligned} \rho_\varepsilon(A) &= \max\{|z| : z \in \Lambda(A+\Delta), \|\Delta\| \leq \varepsilon\} && (1.3a) \\ &= \max\{|z| : z \in \mathbb{C}, \|(zI-A)^{-1}\| \geq \varepsilon^{-1}\}. && (1.3b) \end{aligned}$$

[*] Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106 Germany `mitchell@mpi-magdeburg.mpg.de`. The author's visits to the Courant Institute of Mathematical Sciences, New York University were supported by the U.S. National Science Foundation grant DMS-1620083

For $\varepsilon = 0$, $\rho_\varepsilon(A) = \rho(A)$, the *spectral radius* of A, and so it is easy to see that $\mathcal{K}(A) = \infty$ if $\rho(A) > 1$. Furthermore, if A is normal and $\rho(A) \leq 1$, then $\mathcal{K}(A) = 1$, which is the minimum value $\mathcal{K}(A)$ can take, since k in (1.1) can be zero.

Correspondingly, there is also a continuous-time version of the Kreiss Matrix Theorem that states for any matrix $A \in \mathbb{C}^{n \times n}$ [TE05, Theorem 18.5]

$$\mathcal{K}(A) \leq \sup_{t \geq 0} \|\mathrm{e}^{tA}\| \leq \mathrm{e}n\mathcal{K}(A), \tag{1.4}$$

where this version of $\mathcal{K}(A)$ also has two equivalent definitions [TE05, Eq. 14.7]

$$\begin{aligned} \mathcal{K}(A) &= \sup_{z \in \mathbb{C}, \operatorname{Re} z > 0} (\operatorname{Re} z)\|(zI - A)^{-1}\| && \text{(1.5a)} \\ &= \sup_{\varepsilon > 0} \frac{\alpha_\varepsilon(A)}{\varepsilon}, && \text{(1.5b)} \end{aligned}$$

and the *ε-pseudospectral abscissa* $\alpha_\varepsilon(\cdot)$ is defined by

$$\begin{aligned} \alpha_\varepsilon(A) &= \max\{\operatorname{Re} z : z \in \Lambda(A + \Delta), \|\Delta\| \leq \varepsilon\} && \text{(1.6a)} \\ &= \max\{\operatorname{Re} z : z \in \mathbb{C}, \|(zI - A)^{-1}\| \geq \varepsilon^{-1}\}. && \text{(1.6b)} \end{aligned}$$

If $\varepsilon = 0$, $\alpha_\varepsilon(A) = \alpha(A)$, the *spectral abscissa* of A, and so $\mathcal{K}(A) = \infty$ if $\alpha(A) > 0$. Similar to the discrete-time case, $\mathcal{K}(A) \geq 1$ since t in (1.4) can be zero and $\mathcal{K}(A) = 1$ if A is normal and $\alpha(A) \leq 0$.

To date, Kreiss constants have typically been estimated using supervised techniques, i.e., where a user is an active participant of the process. For example, in [Men06, Chapter 3.4.1] and [EK17], Kreiss constants have been approximated by plotting (1.2b) or (1.5b) and simply taking the maximum of the resulting curve. Kreiss constants have also been estimated by plotting $\|\mathrm{e}^{tA}\|$ with respect to t or $\|A^k\|$ with respect to k, as well as by finding local maximizers of (1.2b) or (1.5b) via optimization [TE05, Chapters 14 and 15]. Plotting techniques of course have low fidelity. They are unlikely to obtain the value of $\mathcal{K}(A)$ to more than a few digits at best and may require a large number of function evaluations to have any accuracy whatsoever. Optimization techniques on the other hand, under sufficient regularity conditions, have high fidelity in finding local maximizers and can often do so with relatively few function evaluations. However, for nonconvex optimization problems like those in (1.7), general optimization solvers cannot guarantee that a global maximizer, and thus $\mathcal{K}(A)$, is obtained. Indeed, none of these techniques can guarantee that $\mathcal{K}(A)$ is computed, as this would require knowing *a priori* an interval that contains a global maximizer of these particular functions, an interval which is also sufficiently sampled (to use plotting) or contains no other stationary points (to use optimization). Of course, such intervals are not necessarily knowable in advance and/or without user involvement. Furthermore, when transient behavior happens on a very fast time scale, suitable intervals would be very small and so may be hard to find, particularly without fine-grained sampling. As noted by Mengi [Men06, Section 6.2.2], "in general it is difficult to guess *a priori* which ε value is most relevant for the transient peak [of (1.2b) or (1.5b)]."

In contrast, we recently proposed the first algorithm to compute $\mathcal{K}(A)$ with theoretical guarantees [Mit19], which was done by working with the inverses of (1.2a) and (1.5a), respectively

$$\begin{aligned} \mathcal{K}(A)^{-1} &= \inf_{|z|>1} \sigma_{\min}\left(\frac{zI - A}{|z| - 1}\right) && \text{(discrete-time)} && \text{(1.7a)} \\ \mathcal{K}(A)^{-1} &= \inf_{\operatorname{Re} z>0} \sigma_{\min}\left(\frac{zI - A}{\operatorname{Re} z}\right) && \text{(continuous-time),} && \text{(1.7b)} \end{aligned}$$

and exploiting the similarity of solving these optimization problems to that of computing the *distance to uncontrollability.* For a linear control system

$$\dot{x} = Ax + Bu, \tag{1.8}$$

where $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, and both the state $x \in \mathbb{C}^n$ and control input $u \in \mathbb{C}^p$ are dependent on time, the distance to uncontrollability $\tau(A, B)$ can be computed by solving [Eis84]

$$\tau(A, B) = \min_{z \in \mathbb{C}} \sigma_{\min} \left(\begin{bmatrix} A - zI & B \end{bmatrix} \right). \tag{1.9}$$

Indeed, the algorithms we proposed in [Mit19] for computing Kreiss constants are inspired by the existing methods of [Gu00, BLO04, GMO+06] for $\tau(A, B)$.

The main drawback of all of these methods is that they rely on a set of related *level-set tests*, which can be used to guarantee convergence to $\tau(A, B)$ or $\mathcal{K}(A)$ but are quite expensive to compute, e.g., $\mathcal{O}(n^6)$ work. These particular tests can also be sensitive to rounding errors, which can cause them to fail in practice. On the other hand, local minimizers of the optimization problems in (1.7) and (1.9) can be found reliably and rather cheaply, relatively speaking, using standard optimization techniques. For finding minimizers which are sufficiently smooth, typically only a handful of optimization iterations are needed, while the dominant cost to evaluate the value, gradient, and Hessian of any of these objective functions amounts to a single SVD, and hence is $\mathcal{O}(n^3)$ work for dense systems. As we noted in [Mit19], standard optimization is also efficient for obtaining locally-optimal approximations to $\tau(A, B)$ and $\mathcal{K}(A)$ when n is large; (1.7) and (1.9) only have two optimization variables and minimum singular values of large matrices, along with their gradients with respect to the two variables, can be computed in $\mathcal{O}(n)$ work via sparse SVD solvers. Indeed, to minimize the number of times the costly level-set tests are invoked, both [BLO04] and [Mit19] proposed optimization-with-restarts methods, where "cheap" optimization is used to find local minimizers (of (1.9) and (1.7), respectively) and the level-set tests are only used to construct *globality certificates*. On each round of optimization, a certificate either asserts that a point is a global minimizer (which terminates the algorithm) or the test provides new points from which optimization can be restarted. In the latter case, the new starting points guarantee that a better (lower) minimizer will be obtained in the next round of optimization. As a result, these two optimization-with-restarts methods respectively converge to $\tau(A, B)$ and $\mathcal{K}(A)$.

While in [Mit19] we answered the open question of how to extend the existing state-of-the-art techniques for computing $\tau(A, B)$ to realize the first algorithm for computing Kreiss constants, computing either quantity nevertheless remains prohibitively expensive for all but the smallest of problems. Motivated by these limitations, in this paper we develop new globality certificates for both $\mathcal{K}(A)$ and $\tau(A, B)$ that are potentially much faster to compute and more reliable numerically, with the purpose of using them in the optimization-with-restarts methods of [Mit19] and [BLO04]. The core idea is to construct a new function that, when sufficiently resolved by an interpolant on a *finite interval known a priori*, can indicate that a given point is a global minimizer of (1.7a), (1.7b), or (1.9) (as appropriate). If the given point is not a global minimizer, then typically a low-fidelity interpolant will suffice to provide new starting points guaranteeing a better minimizer will be found in the subsequent round of optimization. Although our new globality-certificate function may have to be evaluated many times in this process, it only needs $\mathcal{O}(n^3)$ work per evaluation and these evaluations can be done in an embarrassingly parallel manner. As such, we think this may be a more pragmatic approach than the earlier level-set tests discussed above, particularly for larger problems where $\mathcal{O}(n^6)$ work is simply intractable. Furthermore, provided sufficiently accurate interpolants are obtained, our new approach should be more reliable, since it (a) avoids the computations of the earlier level-set tests that are numerically sensitive and (b) can also benefit from appropriate structure-preserving eigenvalue solvers to avoid other numerical issues.

The paper is organized as follows. We first give a high-level overview of the existing methods of [Gu00, BLO04, GMO+06] for $\tau(A, B)$ and [Mit19] for $\mathcal{K}(A)$ and discuss their properties in §2. Then in §3 we present our new interpolation-based globality certificates for the case of computing continuous-time $\mathcal{K}(A)$. Analogues of our interpolation-based certificates for discrete-time $\mathcal{K}(A)$ and $\tau(A, B)$ are respectively derived in §4 and §5. Numerical experiments are given in §6, while concluding remarks are made in §7.

2 Existing methods and their limitations

The following algorithms for $\tau(A,B)$ and $\mathcal{K}(A)$ all trace back to a novel, albeit expensive, level-set test developed by Gu [Gu00, Section 3.2], which Gu used to develop an iteration for estimating $\tau(A,B)$ to within a factor of two [Gu00]. The dominant cost of Gu's algorithm is performing the test itself, which, as originally stated, is $\mathcal{O}(n^6)$ work since it involves computing all eigenvalues of certain $2n^2 \times 2n^2$ matrix pencils.[1] Given two real parameters, specifically a guess $\gamma \geq 0$ for the value of $\tau(A,B)$ and some $\eta \geq 0$, the test determines whether there exists one or more points $z \in \mathbb{C}$ such that

$$\sigma_{\min}\left(\begin{bmatrix} A - zI & B \end{bmatrix}\right) = \sigma_{\min}\left(\begin{bmatrix} A - (z+\eta)I & B \end{bmatrix}\right) = \gamma. \tag{2.1}$$

In other words, the test determines if there exists a pair of points on the γ-level set of $\sigma_{\min}\left(\left[\begin{smallmatrix} A-zI & B \end{smallmatrix}\right]\right)$ that are a distance η apart horizontally. If the test determines there are no such points, it does not imply that $\tau(A,B) > \gamma$ but another key result of [Gu00, Theorem 3.1] shows that $\tau(A,B) > \gamma - \frac{\eta}{2}$ instead holds as a lower bound.

Gu's algorithm was followed by [BLO04], which employed this new test in two different ways to compute $\tau(A,B)$ to a given tolerance, not just estimate it to within a factor of two. The first of these, a linearly-convergent trisection algorithm, maintains a bracketing interval containing $\tau(A,B)$ and uses the test to determine whether the upper or lower third of this interval can be discarded on each iteration. However, the authors of [BLO04] instead recommended using their second proposed algorithm, the optimization-with-restarts method mentioned in the introduction, since it is typically much faster. Though both require $\mathcal{O}(n^6)$ work, since they each use the test of Gu, the optimization-with-restarts method generally invokes it far fewer times than the trisection algorithm. Finally, using a divide-and-conquer technique, [GMO$^+$06] showed how additional structure in the test of Gu could be exploited to reduce its work complexity to $\mathcal{O}(n^4)$ on average and $\mathcal{O}(n^5)$ in the worst case.

Returning to the case of Kreiss constants, the trisection algorithm does not appear to extend to computing $\mathcal{K}(A)$ [Mit19, Section 4], and even if it did extend, it would likely be inaccurate when $\mathcal{K}(A)$ is large [Mit19, Section 4.1]. In fact, the original trisection algorithm for $\tau(A,B)$ may struggle to deliver any accuracy whatsoever, perhaps not even a single digit, if $\tau(A,B)$ is close to zero [Mit19, Section 4.1].

Instead, we proposed two methods [Mit19, Sections 3 and 5] for computing continuous- and discrete-time Kreiss constants via applying the optimization-with-restarts approach to (1.7b) and (1.7a), respectively. The core part of course is to develop the corresponding computable globality certificates which not only assess potential solutions to (1.7) but also provide good starting points for the next round of optimization if a global minimizer has not yet been reached. In [Mit19, Sections 3.2 and 5.2], we accomplished this by developing new continuous- and discrete-time level-set tests for $\mathcal{K}(A)^{-1}$ that were inspired by Gu's test for estimating $\tau(A,B)$ and which also require $\mathcal{O}(n^6)$ work and operate in a similar fashion. In the continuous-time case, letting

$$f(z) \coloneqq \sigma_{\min}\left(\frac{zI - A}{\operatorname{Re} z}\right), \tag{2.2}$$

parameter $\gamma \in [0,1)$ is a guess for the value of $\mathcal{K}(A)^{-1}$, $\eta \geq 0$ is the fixed distance between pairs of points on the γ-level set of $f(z)$, and angle θ sets the required orientation for all such pairs. The new continuous-time $\mathcal{K}(A)^{-1}$ test determines whether there exists one or more points $z \in \mathbb{C}$ such that

$$f(z) = f(z + \eta e^{\mathbf{i}\theta}) = \gamma. \tag{2.3}$$

If the test finds any such points, optimization is guaranteed to find better (lower) minimizers to (1.7b) when restarted from these points. Hence the test is performed for decreasing values of

[1] As in [BLO04], work complexities are given in terms of considering all computations of singular values, eigenvalues, etc., as *atomic operations* with cubic costs in the dimensions of the associated matrices, and we further assume that these costs reduce to linear if sparse methods are applicable.

η, until either one or more points satisfying (2.3) is found and so optimization is restarted, or η falls below a tolerance indicating a global minimizer has been found. While we additionally showed that the asymptotically faster divide-and-conquer method of [GMO$^+$06] does extend to our new $\mathcal{K}(A)^{-1}$ tests derived in [Mit19, Sections 3.3 and 5.3], this appears to only be a theoretical achievement. In practice, we observed that divide-and-conquer in the Kreiss constant setting was unreliable [Mit19, Section 6.3]. Consequently, computing Kreiss constants with any guarantees, i.e., via the globality certificates we derived in [Mit19], seems to be an $\mathcal{O}(n^6)$ affair.

The high cost of these level-set tests limits computing $\tau(A,B)$ and $\mathcal{K}(A)$ to rather small problems. In the Kreiss constant case, the $\mathcal{O}(n^6)$ work complexity results also hide even higher constant terms compared to the test of Gu for $\tau(A,B)$. As mentioned earlier, the original $\tau(A,B)$ test derived by Gu involves computing the eigenvalues of a $2n^2 \times 2n^2$ *generalized* eigenvalue problem. This was later simplified to a $2n^2 \times 2n^2$ *standard* eigenvalue problem in [GMO$^+$06, Section 3.1]. However, our new tests for $\mathcal{K}(A)^{-1}$ require computing eigenvalues of $4n^2 \times 4n^2$ *generalized* eigenvalue problems in the continuous-time case and $4n^2 \times 4n^2$ *quadratic* eigenvalue problems in the discrete-time case. Part of the reason for these differences is that while all these cases involve $4n^2 \times 4n^2$ pencils at some point in their derivations, there is additional structure in the $\tau(A,B)$ test that can be exploited to reduce the problem size down to $2n^2 \times 2n^2$ and from a general to a standard eigenvalue problem; however, this exploitable structure is not present in the Kreiss constant setting [Mit19, Remark 3.2] so it is unclear if any reductions are possible here. Furthermore, the $\mathcal{O}(n^4)$ and $\mathcal{O}(n^5)$ (average and worst case) work complexities of the faster divide-and-conquer technique, which again does not seem to work well for computing Kreiss constants, are still quite limiting, albeit not as severely as $\mathcal{O}(n^6)$.

As mentioned in the introduction, these level-set tests also come with some numerical difficulties, for both $\tau(A,B)$ and Kreiss constants. The first and most problematic issue is that these tests can be quite sensitive to the parameter η. If there exist points satisfying (2.1) (or (2.3)) for the given values of γ and η (or γ, η, and θ), the corresponding tests may, due to rounding errors, fail to return any of them, especially if η is small. Since all of the algorithms above eventually require performing the tests for diminishing values of η, this can lead to inaccuracy. This sensitivity to η is precisely what motivated the trisection algorithm as a more reliable alternative to using bisection to compute $\tau(A,B)$ [BLO04, p. 358]. However, this same sensitivity is ultimately why the trisection algorithm may nevertheless still fail to have any accuracy when $\tau(A,B)$ is small [Mit19, Corollary 4.2].

With the optimization-with-restarts methods, the sensitivity to η plays out a bit differently. In the trisection algorithm, a lower or upper bound is updated on every iteration based on the result of a *single* level-set test using a value of η determined by trisection itself. If sensitivity to η causes the test to incorrectly return no points, the lower bound will be updated erroneously. On the other hand, the globality certificates computed in optimization-with-restarts generally involve computing the level-set tests for multiple values of η before asserting globality and can be initialized with relatively large values for η. As such, it seems reasonable that optimization-with-restarts methods may be more fault-tolerant than the trisection algorithm. Furthermore, even if the globality certificate computation incorrectly asserts a minimizer is a global one, which would terminate the algorithm when a restart should be performed, how much this affects the overall accuracy is unpredictable. The accuracy depends on how close the function values differ between the current local minimizer and a global one. It is plausible the optimization-with-restarts may still find good local minimizers, i.e., that attain objective values close to $\mathcal{K}(A)^{-1}$ or $\tau(A,B)$, particularly if optimization is run from many points on each round.

The second issue is much less critical and only relevant in the Kreiss constant setting. Here the tests additionally require that $\gamma < 1$. The reason for this additional restriction is that the level-set tests look for the presence of a *pair* of points that are a distance η apart. If $\gamma \in [\mathcal{K}(A)^{-1}, 1)$, then there exists an $\eta_{\max} > 0$ such that the tests should detect such pairs for all $\eta \in [0, \eta_{\max}]$ [Mit19, Section 4.2]. However, such pairs may not exist for $\gamma \geq 1$. For example, in the continuous-time case, all lines in the complex plane at height $\gamma \geq 1$ may only intersect the corresponding level set of (2.2) at most once. Hence in [Mit19], we simply assume that the first round of optimization finds at least one minimizer $\tilde{z}$ of (2.2) such that $f(\tilde{z}) < 1$, which by monotonicity of the optimization-with-restarts methods, ensures that $\gamma < 1$ holds for all globality certificate computations; the assumption is also necessary in the discrete-time case, i.e., at least one of the minimizers of (1.7a) found in the first round of optimization must have a corresponding function value that is strictly less than one. This does not seem to be difficult to satisfy in practice but it is an additional complication.

3 Interpolation-based globality certificates for continuous-time $\mathcal{K}(A)$

We begin by noting that parameters γ and η are integral parts of both Gu's original estimation method and the trisection algorithm. In both methods, on each iteration whether $\tau(A,B) > \gamma$ holds is unknown, and so Gu's test is used to verify either $\tau(A,B) \leq \gamma$ or $\tau(A,B) > \gamma - \frac{\eta}{2}$. Meanwhile, parameter η assesses the accuracy at any given iteration, since for sufficiently small η, $\gamma - \frac{\eta}{2} \approx \gamma$. However, in the context of using optimization with restarts, this framework is a bit out of sync with reality. Clearly any locally-optimal solutions to the problems of (1.7) and (1.9) must correspond to a function value γ that is at least $\mathcal{K}(A)^{-1}$ or $\tau(A,B)$, respectively, so there is no need to verify this. Furthermore, how accurately *locally-optimal* solutions to (1.7) and (1.9) are resolved lies with the optimization solver, not with the value of η. This is evident in the globality certificate computations, where it is not so critical to perform the level-set tests with specific values of η; in this context, parameter η is simply an artifact of the original level-set test of Gu that has been repurposed for detecting if the current minimizer is a global one or not.

With this different perspective in mind, we consider abandoning the concept of looking for pairs of points a fixed distance apart specified by η. In its place, we propose constructing a new globality certificate that, given a local minimizer whose function value is γ, answers the question: are there other points on this same level set and if so, where are they? To do this, we will use the following results.

Consider the singular value function in (1.7b) parameterized in polar coordinates:

$$f_{\mathrm{c}}(r,\theta) = \sigma_{\min}(F_{\mathrm{c}}(r,\theta)) \quad \text{and} \quad F_{\mathrm{c}}(r,\theta) = \frac{r\mathrm{e}^{\mathbf{i}\theta}I - A}{r\cos\theta}, \tag{3.1}$$

so

$$\mathcal{K}(A)^{-1} = \inf_{r>0,\, \theta \in (-\frac{\pi}{2},\frac{\pi}{2})} f_{\mathrm{c}}(r,\theta).$$

We assume $\alpha(A) \leq 0$ (as otherwise $\mathcal{K}(A) = \infty$) and that zero is not an eigenvalue of A. The reason for the exclusion of zero as an eigenvalue will become clear momentarily. For a fixed $\theta \in \mathbb{R}$, the following key result relates singular values of $F_{\mathrm{c}}(r,\theta)$ with eigenvalues of a certain $2n \times 2n$ matrix pencil. Exploiting such relationships of singular values and eigenvalues has a rich history in computing various robust stability measures, dating back to at least 1988 when Byers introduced the first algorithm for computing the distance to instability [Bye88].

Theorem 3.1. *Given finite parameters $\gamma, r, \theta \in \mathbb{R}$, with $\gamma \geq 0$ and $r > 0$, then γ is a singular value of $F_{\mathrm{c}}(r,\theta)$ defined in* (3.1) *if and only if r is an eigenvalue of matrix pencil (M, N_θ), where*

$$M := \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix} \quad \textit{and} \quad N_\theta := \begin{bmatrix} -\gamma\cos\theta I & \mathrm{e}^{\mathbf{i}\theta} I \\ \mathrm{e}^{-\mathbf{i}\theta} I & -\gamma\cos\theta I \end{bmatrix} \tag{3.2}$$

are Hermitian matrices and N_θ is indefinite if $|\gamma\cos\theta| < 1$.

Proof. Suppose γ is a singular value of $F_{\mathrm{c}}(r,\theta)$ with left and right singular vectors u and v. Then

$$\left(\frac{r\mathrm{e}^{\mathbf{i}\theta}I - A}{r\cos\theta}\right) v = \gamma u \quad \text{and} \quad \left(\frac{r\mathrm{e}^{-\mathbf{i}\theta}I - A^*}{r\cos\theta}\right) u = \gamma v.$$

Multiplying both equations by $r\cos\theta$ and then rearranging terms yields:

$$Av = r\left(-\gamma\cos\theta u + \mathrm{e}^{\mathbf{i}\theta}v\right) \quad \text{and} \quad -A^*u = r\left(-\mathrm{e}^{-\mathbf{i}\theta}u + \gamma\cos\theta v\right).$$

These two equations can then be written as this generalized eigenvalue problem:

$$\begin{bmatrix} -A^* & 0 \\ 0 & A \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = r\begin{bmatrix} -\mathrm{e}^{-\mathbf{i}\theta}I & \gamma\cos\theta I \\ -\gamma\cos\theta I & \mathrm{e}^{\mathbf{i}\theta}I \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}.$$

Multiplying this generalized eigenvalue problem on the left by the unitary matrix $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$ does not change the spectrum and yields (3.2), thus proving the if-and-only-if equivalence.

Since N_θ is composed of four blocks of different multiples of the $n \times n$ identity matrix, it is easy to see that its eigenvalues are $-\gamma\cos\theta \pm 1$. Hence N_θ must be indefinite if $|\gamma\cos\theta| < 1$. □

Remark 3.2. *If a point $(\tilde{r}, \tilde{\theta})$ is in the γ-level set of $f_{\mathrm{c}}(r,\theta)$ for some $\gamma \geq 0$, then by Theorem 3.1, it follows that $\tilde{r}$ is an eigenvalue of the matrix pencil $(M, N_{\tilde{\theta}})$. Note that the converse is not necessarily true, i.e., if $\tilde{r}$ is an eigenvalue of the matrix pencil $(M, N_{\tilde{\theta}})$, Theorem 3.1 only states that γ is a singular value of $F_{\mathrm{c}}(\tilde{r}, \tilde{\theta})$. For point $(\tilde{r}, \tilde{\theta})$ to also be in the γ-level set, γ would additionally have to be the smallest singular value of $F_{\mathrm{c}}(\tilde{r}, \tilde{\theta})$. However, when γ is not the minimum singular value of $F_{\mathrm{c}}(r,\theta)$, it means that point $(\tilde{r}, \tilde{\theta})$ is instead in some $\hat{\gamma}$-level set of $f_{\mathrm{c}}(r,\theta)$ with $\hat{\gamma} < \gamma$.*

Lemma 3.3. *Given $A \in \mathbb{C}^{n\times n}$, let M be as defined in* (3.2) *and $N \in \mathbb{C}^{2n\times 2n}$ any other matrix. Then the matrix pencil (M, N) has zero as an eigenvalue if and only if matrix A has zero as an eigenvalue.*

Proof. The proof is immediate since if x and y are right and left eigenvectors for the zero eigenvalue of A, then

$$0 = \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}\begin{bmatrix} y \\ x \end{bmatrix} = 0 \cdot N\begin{bmatrix} y \\ x \end{bmatrix}.$$

□

Theorem 3.4. *The spectrum of matrix pencil (M, N_θ) defined by* (3.2) *is symmetric with respect to the real axis and, provided $|\gamma\cos\theta| \neq 1$, is also equivalent to the spectrum of*

$$M_\theta := \frac{1}{(\gamma\cos\theta)^2 - 1}\begin{bmatrix} -\mathrm{e}^{\mathbf{i}\theta}A^* & -\gamma\cos\theta A \\ -\gamma\cos\theta A^* & -\mathrm{e}^{-\mathbf{i}\theta}A \end{bmatrix}. \tag{3.3}$$

Proof. Since $\det(N_\theta) = (\gamma\cos\theta)^2 - 1$, as long as $|\gamma\cos\theta| \neq 1$ holds, N_θ is invertible with inverse

$$N_\theta^{-1} = \frac{1}{(\gamma\cos\theta)^2 - 1}\begin{bmatrix} -\gamma\cos\theta I & -\mathrm{e}^{\mathbf{i}\theta}I \\ -\mathrm{e}^{-\mathbf{i}\theta}I & -\gamma\cos\theta I \end{bmatrix},$$

and so the spectrum of (M, N_θ) can be rewritten as the eigenvalues of $N_\theta^{-1}M$, which is equal to (3.3).

Still assuming that $|\gamma\cos\theta| \neq 1$ for now, as the matrices of (3.2) are both Hermitian, with N_θ invertible, the eigenvalues of the matrix pencil (M, N_θ) are symmetric with respect to the real axis by a result of Lancaster and Ye [LY91, Theorem 2.2]. Since N_θ is only singular for at most four distinct values of $\theta \in [0, 2\pi)$, by continuity of eigenvalues, the eigenvalues of (M, N_θ) remain symmetric with respect to the real axis even if $|\gamma\cos\theta| = 1$, thus completing the proof. □

We are now ready to present our new globality certificate for (1.7b). Given a $\gamma \geq 0$, the idea is to sweep the open right half of the complex plane with rays from the origin to determine which ones intersect the γ-level set. To do this, we will construct a nonnegative continuous function $d_{\mathrm{c}} : (-\frac{\pi}{2}, \frac{\pi}{2}) \to [0, \pi^2]$ such that $d_{\mathrm{c}}(\tilde{\theta}) = 0$ holds if and only if the ray from the origin determined by $\tilde{\theta}$ intersects either the γ-level set of $f_{\mathrm{c}}(r,\theta)$ or, per Remark 3.2, another $\hat{\gamma}$-level set with $\hat{\gamma} < \gamma$. Hence, if $d_{\mathrm{c}}(\theta)$ is strictly positive for all $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, then $\gamma < \mathcal{K}(A)^{-1}$ must hold. Otherwise, the angles $\tilde{\theta}$ for which $d_{\mathrm{c}}(\tilde{\theta}) = 0$ provide the directions of the rays that intersect the γ-level or $\hat{\gamma}$-level sets (with $\hat{\gamma} < \gamma$), and these intersection points can be used to restart optimization in order to find a better minimizer of (1.7b).

Keeping in mind that the spectrum of (M, N_θ) is always real-axis symmetric, to accomplish our criteria above, we use the function

$$d_{\mathrm{c}}(\theta) \coloneqq \min\{\operatorname{Arg}(\lambda)^2 : \lambda \in \Lambda(M, N_\theta), \operatorname{Im}\lambda \geq 0\}, \tag{3.4}$$

where $\operatorname{Arg} : \mathbb{C} \setminus \{0\} \to (-\pi, \pi]$ is the principal value argument function.

Remark 3.5. *Function $d_{\mathrm{c}}(\theta)$ has the following properties:*

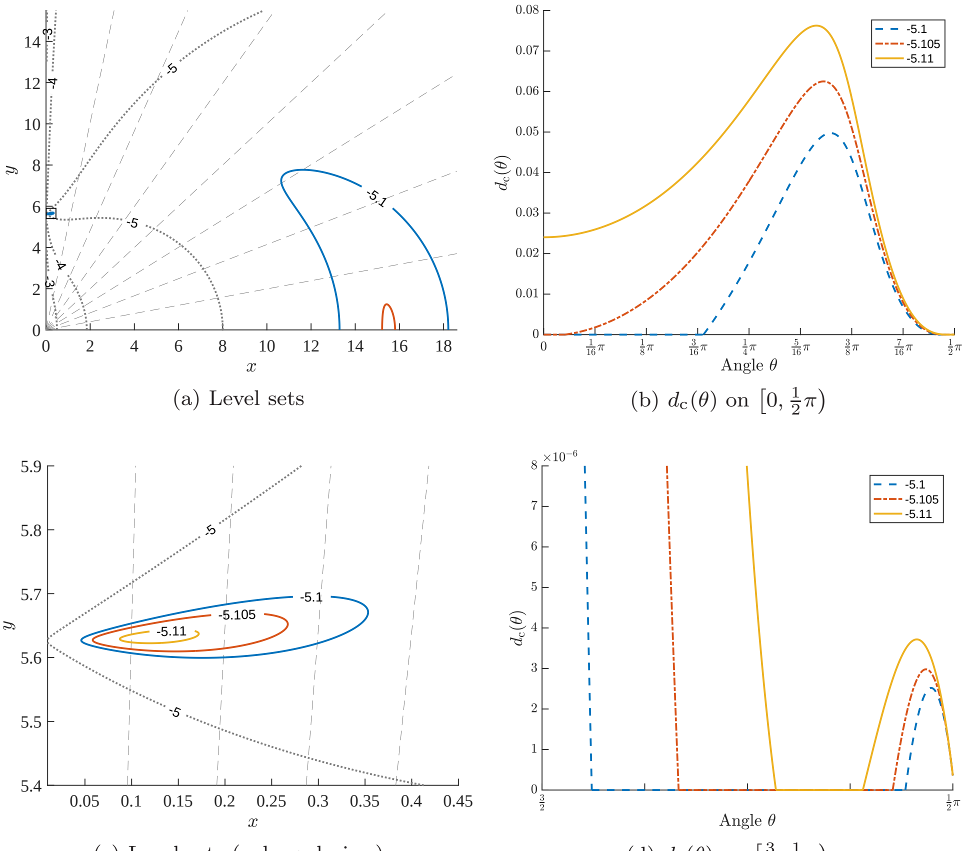
1. $d_{\mathrm{c}}(\theta) \geq 0$ *for all* $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$
2. $d_{\mathrm{c}}(\theta) = 0$ *if and only if* $\Lambda(M, N_\theta)$ *contains an eigenvalue* $r \in \mathbb{R}$, $r > 0$
3. $d_{\mathrm{c}}(\theta)$ *is continuous on its entire domain* $(-\frac{\pi}{2}, \frac{\pi}{2})$
4. $d_{\mathrm{c}}(\theta)$ *is differentiable at a point* θ *if the eigenvalue* $\lambda \in \Lambda(M, N_\theta)$ *attaining the value of* $d_{\mathrm{c}}(\theta)$ *is unique and simple*
5. *If* $\gamma \in (\mathcal{K}(A)^{-1}, 1)$, *then the set* $\mathcal{D} = \{\theta : d_{\mathrm{c}}(\theta) = 0, \theta \in (-\frac{\pi}{2}, \frac{\pi}{2})\}$ *has positive measure.*

The first and second properties hold by construction. The third property is a consequence of the continuity of eigenvalues and our assumption that $0 \notin \Lambda(A)$ and hence by Lemma 3.3, zero is never an eigenvalue of (M, N_θ) for any θ. The fourth property follows from standard perturbation theory for simple eigenvalues and by the definition of $d_{\mathrm{c}}(\theta)$. As we now clarify, the fifth property follows from an argument given in [Mit19, p. 11], which itself is an adaptation of Gu's proof given for [Gu00, Theorem 3.1]. Let $(r_\star, \theta_\star)$ be a global minimizer of (1.7b). If $\gamma \in (\mathcal{K}(A)^{-1}, 1)$, then the γ-level set of $f_{\mathrm{c}}(r,\theta)$ consists of a finite number of continuous closed algebraic curves and $(r_\star, \theta_\star)$ must reside in the interior of one of these curves, which we will call $\mathcal{G}$ and lies entirely in the open right half-plane. Hence there exists an open neighborhood $\mathcal{N}$ about $(r_\star, \theta_\star)$ that lies in the interior of $\mathcal{G}$. Since rays from the origin that sweep through $\mathcal{G}$ must also sweep through $\mathcal{N}$, $d_{\mathrm{c}}(\theta) = 0$ must hold on a set of angles with positive measure.

Taken together, it is clear from Theorem 3.1 and Remark 3.2 that $d_{\mathrm{c}}(\theta)$ meets the criteria outlined above for our new globality certificate. Given $\gamma \geq 0$, $\tilde{r} > 0$ and some $\tilde{\theta} \in (-\frac{\pi}{2}, \frac{\pi}{2})$, if point $(\tilde{r}, \tilde{\theta})$ is in the γ-level set of $f_{\mathrm{c}}(r,\theta)$, then by Theorem 3.1, $\tilde{r}$ must be an eigenvalue of matrix pencil $(M, N_{\tilde{\theta}})$ and so $d_{\mathrm{c}}(\tilde{\theta}) = 0$ holds. If $d_{\mathrm{c}}(\tilde{\theta}) = 0$, by definition there exists a real eigenvalue $\tilde{r} > 0$ of matrix pencil $(M, N_{\tilde{\theta}})$, and so by Theorem 3.1, γ must be a singular value of $F_{\mathrm{c}}(\tilde{r}, \tilde{\theta})$. Thus by Remark 3.2, point $(\tilde{r}, \tilde{\theta})$ must either be in the γ-level of $f_{\mathrm{c}}(r,\theta)$ or some other $\tilde{\gamma}$-level set with $\tilde{\gamma} < \gamma$. Hence, $d_{\mathrm{c}}(\theta) = 0$ is associated with new starting points for optimization such that a better (lower) minimizer can be found. Finally, if $d_{\mathrm{c}}(\theta) > 0$ for all $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, then (M, N_θ) has no positive real eigenvalues for any $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, so again by Theorem 3.1, γ is not a singular value of $F_{\mathrm{c}}(r,\theta)$ for any $r > 0$ and $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$. This in turn means the γ-level set of $f_{\mathrm{c}}(r,\theta)$ is empty. As $f_{\mathrm{c}}(r,\theta)$ is continuous, $\gamma < \mathcal{K}(A)^{-1}$ must hold.

In Figure 1, we show plots of $d_{\mathrm{c}}(\theta)$ for different values of γ for the 10 × 10 continuous-time example used in [Mit19, Section 6.1]. The example is based on a demo from EigTool [Wri02], specifically $A = B - \kappa I$, where $B =$ `companion_demo(10)` and $\kappa = 1.001\alpha(B)$. Since this matrix is real-valued, the level sets of $f_{\mathrm{c}}(r,\theta)$ are symmetric with respect to the real axis and so it is only necessary to sweep the upper right quadrant of the complex plane, i.e., the domain of $d_{\mathrm{c}}(\theta)$ can be reduced to $[0, \frac{\pi}{2})$.

Although we do not know of an analytic way of finding zeros of $d_{\mathrm{c}}(\theta)$, it is a continuous function of one real variable on a fixed finite interval that we can approximate using interpolation. Although $d_{\mathrm{c}}(\theta)$ may be nondifferentiable at some points, modern interpolation software



(a) Level sets

(b) $d_c(\theta)$ on $\left[0, \frac{1}{2}\pi\right)$

(c) Level sets (enlarged view)

(d) $d_c(\theta)$ on $\left[\frac{3}{2}, \frac{1}{2}\pi\right)$

Figure 1: The top left pane shows a contour plot of the level sets (in $\log_{10}$ scale, with label k denoting 10^k) of the objective function in (1.7b) for a continuous-time example, with $z = x + \mathbf{i}y$. As this matrix is real, only the upper right quadrant of the complex plane is shown. The global minimizer of (1.7b) lies in the small boxed area near $(x, y) \approx (0, 6)$; an enlarged view of this region is shown in the bottom left pane. Contours are shown for $k = -3, -4, -5$ (dotted), $k = -5.1$ (solid), $k = -5.105$ (solid, unlabeled at top left), and $k = -5.11$ (solid, not visible at top left). For each of the three solid contours, $d_c(\theta)$ for $\gamma = 10^k$ is plotted in the top and bottom right panes, for the respective regions shown in the top and bottom left panes. For each angle tick mark in the right panes, the corresponding ray from the origin is shown as a dashed line in the top or bottom left pane as appropriate. It is easy to see the correspondence between the level sets for $k \in \{-5.1, -5.105, -5.11\}$ in the left panes and where their associated functions $d_c(\theta)$ are zero in the right panes.

is rather adept at approximating functions that are nonsmooth and even discontinuous or have singularities. Such an interpolation-based approach has several benefits. Provided an appropriate structure-preserving eigensolver is used to preserve the real-axis symmetry of $\Lambda(M, N_\theta)$, it follows that $d_{\mathrm{c}}(\theta)$ will be exactly zero numerically for any θ that corresponds to a ray intersecting the level set. Relatedly, evaluating $d_{\mathrm{c}}(\theta)$ is relatively cheap, since it only requires computing the eigenvalues of a single $2n \times 2n$ eigenvalue problem and different angles can be evaluated simultaneously in an embarrassingly parallel manner. Furthermore, the number of interpolation points needed to build a high-fidelity approximation to $d_{\mathrm{c}}(\theta)$ is not necessarily dependent on n, meaning that is not entirely unreasonable to think that such approximations might be built with only $\mathcal{O}(n^3)$ work, at least for some problems and perhaps with a high constant factor. We can also expect that high-fidelity interpolations will only be needed once minimizers that attain $\mathcal{K}(A)^{-1}$, or close to it, are found. Generically speaking, when $\gamma = \mathcal{K}(A)^{-1}$, we expect that $\mathcal{D}$ has measure zero, but if $\gamma > \mathcal{K}(A)^{-1}$, then $\mathcal{D}$ has positive measure. By building successively better interpolants, which is how many interpolation approximations methods already work, the process can be stopped as soon as any angle yielding $d_{\mathrm{c}}(\theta) = 0$ is encountered. This angle, along with the associated positive real eigenvalues of (3.2), provides the one or more starting points for another round of optimization (provided these points do not happen to be stationary). If $\mathcal{D}$ is relatively large, then the interpolation process should terminate quickly, with a crude approximation. Hence, before $\mathcal{K}(A)^{-1}$ is accurately resolved, high-fidelity approximations to $d_{\mathrm{c}}(\theta)$ may not necessarily be needed. Lastly, as noted earlier, if A is real-valued, then $f_{\mathrm{c}}(r, \theta)$ has real-axis symmetry, so it suffices to approximate $d_{\mathrm{c}}(\theta)$ on the smaller interval $[0, \frac{\pi}{2})$ instead of $(-\frac{\pi}{2}, \frac{\pi}{2})$.

There are some challenges though in approximating $d_{\mathrm{c}}(\theta)$, besides the high number of function evaluations that will likely be needed. First, although continuous, $d_{\mathrm{c}}(\theta)$ may be nondifferentiable at points where there are ties for the minimum value in (3.4). Second, $d_{\mathrm{c}}(\theta)$ may be non-Lipschitz whenever it transitions to or from $d_{\mathrm{c}}(\theta) = 0$. To see this, suppose the γ-level set of $f_{\mathrm{c}}(r, \theta)$ consists of a single continuous closed curve enclosing a nonempty convex interior. Then $\mathcal{D} \subset (-\frac{\pi}{2}, \frac{\pi}{2})$ is simply a single interval and for any θ in the interior of $\mathcal{D}$, $\Lambda(M, N_\theta)$ contains two distinct positive real eigenvalues. However, as θ approaches either end of interval $\mathcal{D}$, this pair coalesces into a single real eigenvalue, after which it may split apart very rapidly when moving off the real axis. Squaring $\mathrm{Arg}(\lambda)$ in (3.4) acts to smooth out this numerically difficult high rate of change.

Remark 3.6. *Note that our interpolation-based globality cerfiticate has two key differences to the supervised techniques discussed in the introduction for estimating Kreiss constants. The first and more important difference is that a global maximizer of* (1.5b) *may be anywhere in* $[0, \infty)$ *and may occur on a very fast time scale, which can make finding such maximizers very difficult. Here,* $d_{\mathrm{c}}(\theta)$ *is defined on* the fixed finite interval $(-\frac{\pi}{2}, \frac{\pi}{2})$, *and its zeros form a subset with* positive measure *when* $\gamma > \mathcal{K}(A)^{-1}$. *Hence finding zeros of* $d_{\mathrm{c}}(\theta)$ *should be substantially easier than finding global maximizers of* (1.5b). *Second,* $d_{\mathrm{c}}(\theta)$ *is more reliable to compute and cheaper to obtain; computing* $\alpha_\varepsilon(A)$ *via the criss-cross algorithms of [BLO03] or [BM17] often involves computing all eigenvalues of several* $2n \times 2n$ *matrices.*

Remark 3.7. *Certainly our certificate function defined in* (3.4) *is not the only possible choice but one might wonder why we did not choose something simpler, e.g., an indicator function. The reason is that if* $d_{\mathrm{c}}(\theta)$ *were to return a fixed positive value whenever the associated ray does not intersect the level set, then interpolation software may erroneously conclude with very few sample points that the function is constant. This is because the error between the interpolant and* $d_{\mathrm{c}}(\theta)$ *would be exactly zero if none of the interpolation points happen to fall in* $\mathcal{D}$, *which may be small when* γ *is close to* $\mathcal{K}(A)^{-1}$. *Defining* $d_{\mathrm{c}}(\theta)$ *so that it generally varies with* θ *helps to ensure that the function is sufficiently sampled.*

4 Interpolation-based globality certificates for discrete-time $\mathcal{K}(A)$

We now adapt our new globality certificates for discrete-time Kreiss constants. Following §3, we parameterize the minimum singular value in (1.7a) using polar coordinates:

$$f_{\mathrm{d}}(r,\theta) = \sigma_{\min}(F_{\mathrm{d}}(r,\theta)) \quad \text{and} \quad F_{\mathrm{d}}(r,\theta) = \frac{r\mathrm{e}^{\mathbf{i}\theta} I - A}{r-1}. \tag{4.1}$$

In the previous section, the definition of $d_{\mathrm{c}}(\theta)$ required that zero was never an eigenvalue of (M, N_θ), which by Lemma 3.3 was equivalent to zero not being an eigenvalue of A. Hence, as long as this held, $d_{\mathrm{c}}(\theta)$ was well defined and continuous. Here we will construct an almost identical distance function, based on the spectrum of a different matrix pencil which also must not have zero as an eigenvalue (to ensure that the distance function is well defined and continuous). As we will see, this in turn will be equivalent to γ^2 not being an eigenvalue of AA^*. We also assume that $\rho(A) \leq 1$, as otherwise $\mathcal{K}(A) = \infty$.

Theorem 4.1. *Given finite parameters $\gamma, r, \theta \in \mathbb{R}$, with $\gamma \geq 0$ and $r > 1$, then γ is a singular value of $F_{\mathrm{d}}(r,\theta)$ defined in (4.1) if and only if r is an eigenvalue of matrix pencil (S, T_θ) where*

$$S := \begin{bmatrix} -\gamma I & A \\ A^* & -\gamma I \end{bmatrix} \quad \text{and} \quad T_\theta := \begin{bmatrix} -\gamma I & \mathrm{e}^{\mathbf{i}\theta} I \\ \mathrm{e}^{-\mathbf{i}\theta} I & -\gamma I \end{bmatrix} \tag{4.2}$$

are Hermitian matrices and T_θ is indefinite if $\gamma < 1$.

Proof. Suppose γ is a singular value of $F_{\mathrm{d}}(r,\theta)$ with left and right singular vectors u and v. Then

$$\left(\frac{r\mathrm{e}^{\mathbf{i}\theta} I - A}{r-1}\right) v = \gamma u \quad \text{and} \quad \left(\frac{r\mathrm{e}^{-\mathbf{i}\theta} I - A^*}{r-1}\right) u = \gamma v.$$

Multiplying both equations by $r - 1$ and then rearranging terms yields:

$$-\gamma u + Av = r\left(-\gamma u + \mathrm{e}^{\mathbf{i}\theta} v\right) \quad \text{and} \quad -A^* u + \gamma v = r\left(-\mathrm{e}^{-\mathbf{i}\theta} u + \gamma v\right).$$

These two equations can then be written as this generalized eigenvalue problem:

$$\begin{bmatrix} -A^* & \gamma I \\ -\gamma I & A \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = r \begin{bmatrix} -\mathrm{e}^{-\mathbf{i}\theta} I & \gamma I \\ -\gamma I & \mathrm{e}^{\mathbf{i}\theta} I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

Multiplying on the left by $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$ completes the if-and-only-if equivalence.

Since T_θ is composed of four blocks of different multiples of the $n \times n$ identity, it has eigenvalues $-\gamma \pm 1$ and so T_θ must be indefinite if $\gamma < 1$. □

Note that the point of Remark 3.2, with appropriate substitutions, similarly applies to Theorem 4.1, $f_{\mathrm{d}}(r,\theta)$, and $F_{\mathrm{d}}(r,\theta)$.

Lemma 4.2. *For any $\theta \in \mathbb{R}$ and $\gamma \geq 0$, the matrix pencil (S, T_θ) defined by (4.2) has zero as an eigenvalue if and only if matrix AA^* has γ^2 as an eigenvalue.*

Proof. We can assume $\gamma > 0$ as the $\gamma = 0$ case is a direct consequence of Lemma 3.3. Suppose $0 \in \Lambda(S, T_\theta)$ with right and left eigenvectors x and y, hence

$$\begin{bmatrix} -\gamma I & A \\ A^* & \gamma I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = 0.$$

By the bottom row, $x = \frac{1}{\gamma} A^* y$ and so substituting this into the top row yields

$$-\gamma y + A\left(\tfrac{1}{\gamma} A^* y\right) = 0 \quad \Longleftrightarrow \quad AA^* y = \gamma^2 y,$$

completing this direction of the proof. The other direction holds by following the same steps in reverse. □

Theorem 4.3. *The spectrum of matrix pencil (S, T_θ) defined by (4.2) is symmetric with respect to the real axis and, provided $\gamma \neq 1$, is also equivalent to the spectrum of*

$$S_\theta := \frac{1}{\gamma^2 - 1} \begin{bmatrix} \gamma^2 I - \mathrm{e}^{\mathbf{i}\theta} A^* & \gamma(\mathrm{e}^{\mathbf{i}\theta} I - A) \\ \gamma(\mathrm{e}^{-\mathbf{i}\theta} I - A^*) & \gamma^2 I - \mathrm{e}^{-\mathbf{i}\theta} A \end{bmatrix}. \tag{4.3}$$

Proof. Since T_θ is composed of four blocks of different multiples of the $n \times n$ identity matrix, it has the following inverse

$$T_\theta^{-1} = \frac{1}{\gamma^2 - 1} \begin{bmatrix} -\gamma I & -\mathrm{e}^{\mathbf{i}\theta} I \\ -\mathrm{e}^{-\mathbf{i}\theta} I & -\gamma I \end{bmatrix},$$

provided that $\gamma \neq 1$, and so the spectrum of (S, T_θ) can be rewritten as the eigenvalues of $T_\theta^{-1} S$, which is equal to (4.3). The symmetry of the spectrum of (S, T_θ) with respect to the real axis follows the same argument given in the proof of Theorem 3.4. □

For our interpolation-based globality certificate to (1.7a), we will reuse the idea of sweeping the complex plane with a ray from the origin to see where it intersects the γ-level set of $f_\mathrm{d}(r, \theta)$. Since discrete-time Kreiss constants require that we look for intersections anywhere outside the closed unit disk, not just in the right half-plane, we will construct a new nonnegative function $d_\mathrm{d} : (-\pi, \pi] \to [0, \pi^2]$, quite similar to (3.4) though with a larger domain. Of course, when matrix A is real valued, the level sets of $f_\mathrm{d}(r, \theta)$ are also symmetric with respect to the real axis, in which case the domain of $d_\mathrm{d}(\theta)$ can be reduced to $[0, \pi]$.

To define our discrete-time $\mathcal{K}(A)^{-1}$ analogue of (3.4), for a given finite $\gamma \geq 0$ we use the function:

$$d_\mathrm{d}(\theta) := \min\{\operatorname{Arg}(\lambda)^2 : \lambda \in \Lambda(S, T_\theta) \setminus [0, 1], \operatorname{Im} \lambda \geq 0\}, \tag{4.4}$$

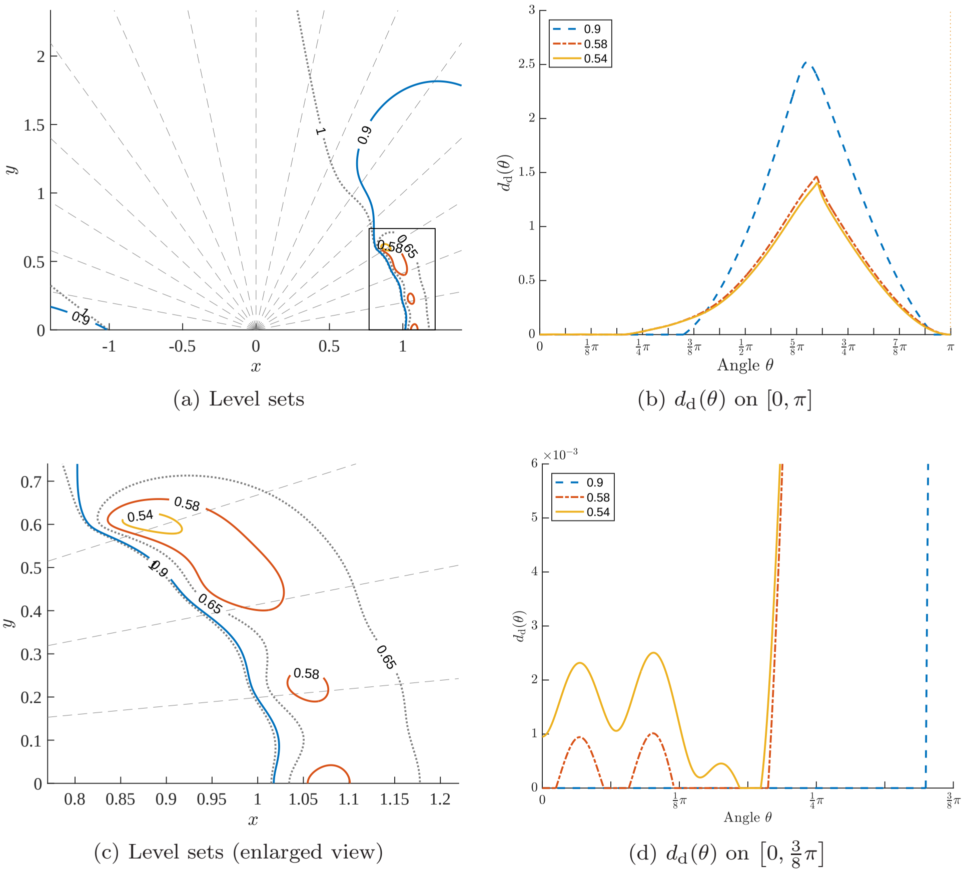
similarly keeping in mind that $\Lambda(S, T_\theta)$ always has real-axis symmetry, regardless of whether or not $f_\mathrm{d}(r, \theta)$ does.

Remark 4.4. *Function $d_\mathrm{d}(\theta)$ has the following properties:*

1. *$d_\mathrm{d}(\theta) \geq 0$ for all $\theta \in (-\pi, \pi]$*
2. *$d_\mathrm{d}(\theta) = 0$ if and only if $\Lambda(S, T_\theta)$ contains an eigenvalue $r \in \mathbb{R}$, $r > 1$*
3. *$d_\mathrm{d}(\theta)$ is continuous on its entire domain $(-\pi, \pi]$*
4. *$d_\mathrm{d}(\theta)$ is differentiable at a point θ if the eigenvalue $\lambda \in \Lambda(S, T_\theta)$ attaining the value of $d_\mathrm{d}(\theta)$ is unique and simple*
5. *If $\gamma \in (\mathcal{K}(A)^{-1}, 1)$, then the set $\mathcal{D} = \{\theta : d_\mathrm{d}(\theta) = 0, \theta \in (-\pi, \pi]\}$ has positive measure.*

The properties of $d_\mathrm{d}(\theta)$ are essentially the same as those of $d_\mathrm{c}(\theta)$ and mostly follow for similar reasons, except for a couple of important differences. The second property holds specifically due to the exclusion of any real-valued eigenvalues of $\Lambda(S, T_\theta)$ that are also in the interval $[0, 1]$ on the real axis. This key change keeps $d_\mathrm{d}(\theta)$ strictly positive whenever $\Lambda(S, T_\theta)$ has one or more eigenvalues in $[0, 1]$ on the real axis but not on $(1, \infty)$. The continuity property is unaffected by this exclusion but does require our assumption that $\gamma^2 \notin \Lambda(AA^*)$, which by Lemma 4.2 guarantees that zero is never an eigenvalue of (S, T_θ) for any $\theta \in \mathbb{R}$. The fifth property follows from a similar argument to the one given for the fifth property of $d_\mathrm{c}(\theta)$.

The removal of eigenvalues of $\Lambda(S, T_\theta)$ in $[0, 1]$ in the definition $d_\mathrm{d}(\theta)$ requires some further comments. In fact, we do not care about any of the eigenvalues of $\Lambda(S, T_\theta)$ that are in the closed unit disk, since by Theorem 4.1 these eigenvalues are only associated with points in level sets of $f_\mathrm{d}(r, \theta)$ that are also in the closed unit disk. However, excluding these eigenvalues could introduce discontinuities whenever an eigenvalue of $\Lambda(S, T_\theta)$ enters or exits the unit disk. Hence, by only excluding $[0, 1]$, continuity is preserved and $d_\mathrm{d}(\theta)$ remains strictly positive if there are no points in the γ-level set of $f_\mathrm{d}(r, \theta)$ outside the unit disk, even though in this case $d_\mathrm{d}(\theta)$ may still



(a) Level sets

(b) $d_\mathrm{d}(\theta)$ on $[0, \pi]$

(c) Level sets (enlarged view)

(d) $d_\mathrm{d}(\theta)$ on $\left[0, \frac{3}{8}\pi\right]$

Figure 2: The top left pane shows a contour plot of the level sets (now in linear scale) of the objective function in (1.7a) for a discrete-time example, with $z = x + \mathbf{i}y$. As this matrix is real, only the upper half of the complex plane is shown. The global minimizer of (1.7a) lies in the boxed area, in the well near the top left corner; an enlarged view of this region is shown in the bottom left pane. Contours are shown for $\gamma = 1$ (dotted), $\gamma = 0.9$ (solid), $\gamma = 0.65$ (dotted), $\gamma = 0.58$ (solid), and $\gamma = 0.54$ (solid, not visible at top left). For each of the solid contours, the corresponding $d_\mathrm{d}(\theta)$ function is plotted in the top and bottom right panes, for the respective regions shown in the top and bottom left panes. For each angle tick mark in the right panes, the corresponding ray from the origin is shown as a dashed line in the left panes. Again the correspondence between the γ-level sets and where their associated functions $d_\mathrm{d}(\theta)$ are zero is clearly evident. In the top right pane, the discontinuity in $d_\mathrm{d}(\theta)$ for $\gamma = 0.54$ near $\theta = \pi$ is due to excluding eigenvalues of (S, T_θ) in an eccentric ellipse centered at the origin.

be infinitesimally small (since eigenvalues of $\Lambda(S, T_\theta)$ can be arbitrarily close to the $[0, 1]$ interval on the real axis). Fortunately, if a structure-preserving eigensolver is used, removing eigenvalues of (S, T_θ) that lie in $[0, 1]$ on the real axis can be done precisely. When a structure-preserving eigensolver is not used, generally there will be rounding errors in the imaginary parts of computed eigenvalues, and so we instead discard any eigenvalue of (S, T_θ) that lies inside an ellipse centered at the origin with major axis 1 and minor axis δ for some small $\delta > 0$. Technically, this may still introduce discontinuities in $d_{\rm d}(\theta)$ but they are much less likely to occur than when excluding the entire unit circle ($\delta = 1$). Such a discontinuity (for $\delta = 10^{-8}$) can be seen in Figure 2, where we show plots of $d_{\rm d}(\theta)$ for different values of γ for the 10×10 discrete-time example used in [Mit19, Section 6.2], namely $A = \frac{1}{13}B + \frac{11}{10}I$, where $B =$ `convdiff_demo(11)` from EigTool.

5 Interpolation-based globality certificates for $\tau(A, B)$

Finally, we adapt our new globality certificates for the distance to uncontrollability, now parameterizing the minimum singular value function in (1.9) using polar coordinates:

$$f_\tau(r, \theta) = \sigma_{\min}(F_\tau(r, \theta)) \quad \text{and} \quad F_\tau(r, \theta) = \begin{bmatrix} A - r\mathrm{e}^{\mathbf{i}\theta} I & B \end{bmatrix}. \tag{5.1}$$

We will again create a distance function based on the spectrum of a certain matrix pencil parameterized by angle θ. As before, it will be necessary to exclude zero as an eigenvalue of this matrix pencil. Here, in the case of $\tau(A, B)$, this will turn out to be equivalent to assuming that γ^2 is not an eigenvalue of $AA^* + BB^*$.

Theorem 5.1. *Given finite parameters $\gamma, r, \theta \in \mathbb{R}$, with $\gamma > 0$ and $r \geq 0$, then γ is a singular value of $F_\tau(r, \theta)$ defined in* (5.1) *if and only if r is an eigenvalue of matrix pencil (G, H_θ) where*

$$G := \begin{bmatrix} \widetilde{B} & A \\ A^* & -\gamma I \end{bmatrix} \quad \text{and} \quad H_\theta := \begin{bmatrix} 0 & \mathrm{e}^{\mathbf{i}\theta} I \\ \mathrm{e}^{-\mathbf{i}\theta} I & 0 \end{bmatrix} \tag{5.2}$$

are Hermitian matrices, H_θ is indefinite, and $\widetilde{B} := \frac{1}{\gamma}BB^* - \gamma I$.

Proof. Suppose γ is a singular value of $F_\tau(r, \theta)$ with left and right singular vectors u and $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$. Then

$$\begin{bmatrix} A - r\mathrm{e}^{\mathbf{i}\theta} I & B \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \gamma u \quad \text{and} \quad \begin{bmatrix} A^* - r\mathrm{e}^{-\mathbf{i}\theta} I \\ B^* \end{bmatrix} u = \gamma \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

From the lower block row of the equation on the right, it is seen that $v_2 = \frac{1}{\gamma}B^* u$. Substituting this into the equation on the left and then, for both this equation on the left and the top block row of the equation on the right, multiplying and rearranging terms respectively yields:

$$\widetilde{B}u + Av_1 = r\left(\mathrm{e}^{\mathbf{i}\theta} v_1\right) \quad \text{and} \quad -A^* u + \gamma v_1 = r\left(-\mathrm{e}^{-\mathbf{i}\theta} u\right),$$

where $\widetilde{B} = \frac{1}{\gamma}BB^* - \gamma I$. These two equations can then be written as this generalized eigenvalue problem:

$$\begin{bmatrix} -A^* & \gamma I \\ \widetilde{B} & A \end{bmatrix} \begin{bmatrix} u \\ v_1 \end{bmatrix} = r \begin{bmatrix} -\mathrm{e}^{-\mathbf{i}\theta} I & 0 \\ 0 & \mathrm{e}^{\mathbf{i}\theta} I \end{bmatrix} \begin{bmatrix} u \\ v_1 \end{bmatrix}.$$

Multiplying on the left by $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$ completes the if-and-only-if equivalence.

It is easy to see that eigenvalues of H_θ are $+1$ and -1, and hence H_θ is indefinite for all θ. □

Note that the above result is essentially a special case, i.e., through the origin, of the level-set search on a straight line derived in [Gu00, Section 2.1]. As Gu noted, similar results were previously developed in [Bye90, Theorem 3.1] and [GN93, Lemmas 2.1 and 2.2]. The point of Remark 3.2, with appropriate substitutions, also applies to Theorem 5.1, $f_\tau(r, \theta)$, and $F_\tau(r, \theta)$.

Lemma 5.2. *Let $\gamma > 0$. Then for any $\theta \in \mathbb{R}$, the matrix pencil (G, H_θ) defined by* (5.2) *has zero as an eigenvalue if and only if matrix $AA^* + BB^*$ has γ^2 as an eigenvalue.*

Proof. Suppose γ^2 is an eigenvalue of $AA^* + BB^*$ with eigenvector x. Then

$$\begin{aligned}(AA^* + BB^*)x = \gamma^2 x \quad &\iff \quad (AA^* + BB^* - \gamma^2 I)x = 0\\ &\iff \quad \tfrac{1}{\gamma}(AA^* + BB^* - \gamma^2 I)x = 0\\ &\iff \quad (\tfrac{1}{\gamma}AA^* + \widetilde{B})x = 0.\end{aligned}$$

Then setting $y = \frac{1}{\gamma}A^*x$, it follows that

$$Ay + \widetilde{B}x = 0 \qquad \text{and} \qquad A^*x - \gamma y = 0,$$

which can be rewritten as

$$\begin{bmatrix}\widetilde{B} & A\\ A^* & -\gamma I\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = 0.$$

Hence, zero is an eigenvalue of the matrix above and so by Lemma 3.3, zero is an eigenvalue of (G, H_θ). The reverse implication holds since all steps are if-and-only-if equivalences. □

Theorem 5.3. *The spectrum of matrix pencil (G, H_θ) defined by* (5.2) *is symmetric with respect to the real axis and is equivalent to the spectrum of*

$$G_\theta := \begin{bmatrix}\mathrm{e}^{\mathbf{i}\theta}A^* & -\gamma\mathrm{e}^{\mathbf{i}\theta}I\\ \mathrm{e}^{-\mathbf{i}\theta}\widetilde{B} & \mathrm{e}^{-\mathbf{i}\theta}A\end{bmatrix}. \tag{5.3}$$

Proof. Clearly $H_\theta = H_\theta^{-1}$ for any value of θ and so the spectrum of (G, H_θ) can be rewritten as the eigenvalues of $H_\theta^{-1}G$, which is equal to (5.3). The symmetry of the spectrum of (G, H_θ) with respect to the real axis holds by [LY91, Theorem 2.2] since G and H_θ are Hermitian and H_θ is always invertible. □

Our interpolation-based globality certificate for (1.9) will need to find intersections with the γ-level set of $\sigma_{\min}([\,A-zI\ \ B\,])$ and, unlike for Kreiss constants, there are now no domain restrictions for where z may lie. Hence we will sweep the entire complex plane with rays from the origin via a distance function $d_\tau : (-\pi, \pi] \to [0, \pi^2]$ quite similar to $d_{\mathrm{c}}(\theta)$ and $d_{\mathrm{d}}(\theta)$. The γ-level set of $\sigma_{\min}([\,A-zI\ \ B\,])$ is symmetric with respect to the real axis if either A and B are both real matrices or A is Hermitian; see Appendix A for the proofs. If either of these cases hold, then the domain can be reduced to $[0, \pi]$.

Given a candidate solution to (1.9), with function value $\gamma > 0$, we apply our interpolation strategy to

$$d_\tau(\theta) := \min\{\mathrm{Arg}(\lambda)^2 : \lambda \in \Lambda(G, H_\theta), \operatorname{Im}\lambda \geq 0\} \tag{5.4}$$

in order to compute globality certificates for $\tau(A, B)$. Note that $\Lambda(G, H_\theta)$ always has real-axis symmetry, even when $\sigma_{\min}([\,A-zI\ \ B\,])$ does not.

Remark 5.4. *Function $d_\tau(\theta)$ has the following properties:*

1. *$d_\tau(\theta) \geq 0$ for all $\theta \in (-\pi, \pi]$*

2. *$d_\tau(\theta) = 0$ if and only if $\Lambda(G, H_\theta)$ contains an eigenvalue $r \in \mathbb{R}$, $r > 0$*

3. *$d_\tau(\theta)$ is continuous on its entire domain $(-\pi, \pi]$*

4. *$d_\tau(\theta)$ is differentiable at a point θ if the eigenvalue $\lambda \in \Lambda(G, H_\theta)$ attaining the value of $d_\tau(\theta)$ is unique and simple*

5. *If $\gamma > \tau(A, B)$, then the set $\mathcal{D} = \{\theta : d_\tau(\theta) = 0,\ \theta \in (-\pi, \pi]\}$ has positive measure.*

The general properties of $d_\tau(\theta)$ remain the same as those of $d_{\rm c}(\theta)$ and $d_{\rm d}(\theta)$ and again follow for similar reasons, except now the third property requires our assumption that γ^2 is not an eigenvalue of $AA^* + BB^*$, which by Lemma 5.2 ensures that zero is never an eigenvalue of (G, H_θ) for any $\theta \in \mathbb{R}$. The fifth property holds by a similar argument to those given earlier for the analogous properties of $d_{\rm c}(\theta)$ and $d_{\rm d}(\theta)$, which traces back to Gu's proof of [Gu00, Theorem 3.1].

Note that by Lemma 5.2 and Theorem 5.1, our assumption that γ^2 is not an eigenvalue of $AA^* + BB^*$ is equivalent to γ not being a singular value of $F_\tau(0,\theta)$ for any $\theta \in \mathbb{R}$. As such, the properties of $d_\tau(\theta)$ hold as long as $\gamma < f_\tau(0,\theta)$. Since optimization-with-restarts monotonically decreases the value of γ until it converges to $\tau(A,B)$, we can guarantee that γ^2 is never an eigenvalue of $AA^* + BB^*$ by initializing at the origin. Provided the origin is not a stationary point, optimization guarantees finding a point $(\tilde{r}, \tilde{\theta})$ such that $f_\tau(\tilde{r}, \tilde{\theta}) < f_\tau(0,\theta)$. Otherwise, either other starting points can be evaluated in order to find a function value lower than $f_\tau(0,\theta)$ or the initial value of γ can simply be set to slightly less than $f_\tau(0,\theta)$ and then a globality check can be done. Finally, although $d_\tau(\theta)$ is not defined for $\gamma = 0$, this is not a problem as there is no need to do a globality check when $f_\tau(r,\theta) = 0$, as $f_\tau(r,\theta)$ is never negative.

For brevity, we forgo showing illustrative plots of $d_\tau(\theta)$ here, but an example is shown later in Figure 3c.

6 Numerical experiments

Recall that the optimization-with-restarts methods for computing Kreiss constants [Mit19] and the distance to uncontrollability [BLO04, GMO$^+$06] all work by using optimization to find a (usually) locally minimal value γ_k of the objective function in (1.7b), (1.7a), or (1.9), as appropriate, and then doing a corresponding expensive level-set test, which either asserts that γ_k is in fact globally minimal or provides new starting points for the $(k+1)$th round of optimization. Fast local optimization uses gradients, and optionally, also Hessians, which for (1.7b) (in Cartesian coordinates) and (1.7a) (in polar coordinates) for Kreiss constants were respectively derived in [Mit19, Sections 3.1 and 5.1]. For $\tau(A,B)$, the gradient for (1.9) (in Cartesian coordinates) is given in [BLO04, p. 358]; its Hessian can be derived in a similar way to that of the Hessian for (1.7b) [Mit19, Section 3.1].

Since our new interpolation-based globality certificates are also intended to be used within optimization-with-restarts methods, we need simply replace the older expensive certificate tests with our hopefully cheaper tests to do a comparison. For the optimization phases, we used `fminunc` from Optimization Toolbox in MATLAB and provided it with both gradients and Hessians. In fact, we reused the same code we used for the numerical experiments in [Mit19], which only required adding in support to also find minimizers of (1.9) and, of course, implementing our new certificates. While all of our interpolation-based certificates use polar coordinates, optimization for (1.7b) and (1.9) is still done using Cartesian coordinates. Optimization-with-restarts terminates when either the globality certificate asserts a global minimizer has been found or if optimization can no longer make meaningful progress. This latter condition is necessary in practice since optimization software will generally not compute minimizers exactly and so the level-set certificate tests may return new starting points even when a global minimizer has been found to numerical precision.

To build interpolant approximations to $d_{\rm c}(\theta)$, $d_{\rm d}(\theta)$, and $d_\tau(\theta)$, we used Chebfun (a recent build, commit `51b3f94`), partly for its sophistication and efficiency in "computing with functions to about 15-digit accuracy"[2] and partly because it is also adept at handling nonsmooth functions when its `splitting` option is enabled. Besides enabling `splitting`, we also set `novectorcheck` as our routines for computing $d_{\rm c}(\theta)$, $d_{\rm d}(\theta)$, and $d_\tau(\theta)$ allow values of θ to be provided as a vector rather than one at a time (in §6.2, we will discuss using parallel processing to evaluate the function being approximated). Now restricting to the continuous-time Kreiss setting for concreteness and

[2] The quote is taken from the homepage of `http://www.chebfun.org`.

clarity, for a given γ_k, our globality certificate commences building a chebfun (the interpolant approximation) of $d_{\mathrm{c}}(\theta)$. However, if $d_{\mathrm{c}}(\theta)$ happens to be zero at one or more values of θ provided by Chebfun, then Chebfun is immediately terminated (by throwing and catching an error) and another round of optimization is done. (Technically, a robust implementation should additionally check that these new starting points are not stationary, or nearly so, before deciding to halt Chebfun early.) Otherwise, Chebfun runs until its default termination criteria are met. In this case, none of the interpolation points are zeros of $d_{\mathrm{c}}(\theta)$, but as a final check, we also compute the intervals, if any, where the interpolant is negative (which is trivial to do for a chebfun). If any such intervals exist, the value of $d_{\mathrm{c}}(\theta)$ is computed at each of their midpoints and any that happen to be zeros of $d_{\mathrm{c}}(\theta)$ are used to restart optimization. If there are no such zeros, globality of γ_k is asserted. The same procedure is done for $d_{\mathrm{d}}(\theta)$ and $d_\tau(\theta)$.

Ideally we would use a structure-preserving eigensolver to compute the eigenvalues of the matrix pencils given by (3.2), (4.2), and (5.2), so that the real symmetry of their spectra would be preserved numerically. In floating point computation, this would guarantee that zeros of $d_{\mathrm{c}}(\theta)$, $d_{\mathrm{d}}(\theta)$, and $d_\tau(\theta)$ remain exactly zero and that $d_{\mathrm{d}}(\theta)$ also remains continuous. In 2004, Mehl in fact proposed such a structure-preserving solver [Meh04] for indefinite generalized Hermitian eigenvalue problems, the same kind as our matrix pencils here. However, motivated by different applications, Mehl's solver assumes that the matrix pencils have no real-valued eigenvalues, which are precisely the eigenvalues of interest in our globality certificates. Unfortunately, we are currently unaware of any other structure-preserving eigensolver for this problem class that would also be suitable for our application here. Instead, we just computed the eigenvalues of the related standard eigenvalue problems given by (3.3), (4.3), and (5.3) using `eig` in MATLAB. To account for rounding errors with this approach, the imaginary part of any computed eigenvalue was set to zero if the magnitude of the imaginary part was no more than 10^{-8}. This means that our routines implementing $d_{\mathrm{c}}(\theta)$, $d_{\mathrm{d}}(\theta)$, and $d_\tau(\theta)$ technically have small discontinuities when transitioning to/from zero. Furthermore, when computing $d_{\mathrm{d}}(\theta)$, we set $\delta = 10^{-8}$ for discarding any eigenvalues of (S, T_θ) that are also in the corresponding eccentric ellipse centered at the origin.

As in [Mit19], our prototype codes implemented in MATLAB are only proof-of-concepts and are tuned not for efficiency but so that multiple restarts are likely to be needed, to better illustrate how our interpolation-based globality certificates work. We plan to add optimized robust implementations for general use to a future release of ROSTAPACK: RObust STAbility PACKage [Mit], an open-source library implemented in MATLAB and licensed under the AGPL. The codes used for the experiments here are included in the supplementary materials. All experiments were done in MATLAB R2017b on a dual-socket compute node (from the `mechthild` cluster at MPI Magdeburg) with two Intel Xeon Gold 6130 processors (16 cores each, 32 total), 192GB of RAM, and CentOS Linux 7.

6.1 Comparisons to earlier methods

Since we will address parallel computation in §6.2, here we consider a single-core evaluation of all the methods. We did this by calling `parpool(1)` in MATLAB and by not using any `parfor` loops.

We begin by comparing our new method for computing continuous-time Kreiss constants with our earlier method [Mit19, Section 3], using three test problems of different sizes; see the first three rows of Table 1. The first, `companion` (stab.), is the stabilized EigTool example matrix we used for Figure 1. The two larger problems, `boeing('S')` and `orrsommerfeld`, are from EigTool.[3] We specifically chose starting points such that at least one restart would be necessary; both methods used these same initial points. For `companion` (stab.), the relative difference between the estimates for $\mathcal{K}(A)$ computed by our new and old method was 1.4×10^{-10}, with our new method returning the slightly worse (lower) estimate. However, this small difference was

[3] The `transient` demo from EigTool, although designed to have transient behavior in both $\|\mathrm{e}^{tA}\|$ and $\|A^k\|$, particularly as n increases, appears to only have one minimizer so we excluded it from consideration here.

				Time (sec.)	
Problem	n	z_0	Computed Value	New	Other
$\mathcal{K}(A)$ (continuous)					
`companion` (stab.)	10	6+6**i**	$1.291867070095026 \times 10^{5}$	0.6	2.0
`boeing('S')`	55	1+50**i**	$3.625410525376937 \times 10^{4}$	334.4	6120.9^{*}
`orrsommerfeld`	100	10+10**i**	$3.932304742813434 \times 10^{1}$	127.6	171841.4^{*}
$\mathcal{K}(A)$ (discrete)					
`convdiff` (mod.)	10	−1+1**i**	$1.895013390905799 \times 10^{0}$	0.2	5.9
`randn` #1 (stab.)	50	1+1**i**	$1.758436065783109 \times 10^{0}$	127.2	3156.9^{*}
`randn` #2 (stab.)	100	1−1**i**	$2.358494955746503 \times 10^{0}$	1197.4	`out-of-mem`*
$\tau(A,B)$					
`kahan` ($m = 20$)	60	0+0**i**	$3.882115122611607 \times 10^{-2}$	2.5	249.5
`kahan` ($m = 30$)	150	0+0**i**	$1.825814695301203 \times 10^{-2}$	720.6	27318.7

Table 1: The eight problems tested. The size of the matrix A is given by n, while z_0 is the initial point used by our new methods (and when relevant, the other methods in the comparison). The corresponding estimates for $\mathcal{K}(A)$ or $\tau(A,B)$ computed by our new methods are given under "Computed Value". The elapsed wall-clock times (in seconds) are given in the two rightmost columns, with the total running times of our new methods shown under "New". The "Other" column gives the running times of our earlier methods [Mit19] for computing continuous- and discrete-time $\mathcal{K}(A)$ or the divide-and-conquer-based $\tau(A,B)$ algorithm of [GMO$^+$06], as appropriate. However, the times marked with an asterisk denote where the full optimization-with-restarts methods of [Mit19] were not tested, due to the prohibitive $\mathcal{O}(n^6)$ cost of their level-set tests, particularly since several are often needed; instead, for these problems, only the time to perform a single one of the relevant expensive tests was recorded. For `randn` #2 (stab.), the running time could not be recorded as attempting to solve a quadratic eigenvalue problem of order $4n^2$ quickly caused an out-of-memory error.

merely due to `fminunc` terminating at slightly different points near the global minimizer; as such, we believe this is simply rounding error, but it is also possible that a different optimization code might be better at obtaining minimizers of (1.7b) to a more consistent level of accuracy. In terms of running times, we see that even for $n = 10$, optimization with our new certificates is faster than that of our older method. Due to the larger sizes of `boeing('S')` and `orrsommerfeld`, it was impractical to run the older algorithm of [Mit19, Section 3]. Instead, for each of these two problems, we only recorded a fraction of its running time, namely the time for `eig` to compute all the eigenvalues of a single instance of the $4n^2 \times 4n^2$ generalized eigenvalue problem [Mit19, Equation (3.11)], using the final value of γ_k computed by our new method. This is generally a small subset of the total computations needed in the full algorithm of [Mit19, Section 3], as often several (e.g., ten or more) different instances of these large eigenvalue problems are solved before termination. Hence, the actual performance gaps between our new and old methods on these two examples are likely to be at least another order magnitude wider than the timings (marked with asterisks in Table 1) would suggest. Nevertheless, we still see that our new method is respectively 18.3 and 1346.7 times faster for `boeing('S')` and `orrsommerfeld`. In Table 2, we show the number of points at which Chebfun evaluates $d_{\mathrm{c}}(\theta)$ for each certificate computation. Before a global minimizer is obtained, generally relatively few values of θ are evaluated by Chebfun before new starting points are discovered and optimization commences again. Furthermore, as hoped, we see that the number of function evaluations needed to build the final interpolants asserting globality seems to be uncorrelated with the size of A. In Figure 3a, we show $d_{\mathrm{c}}(\theta)$ for the final value of γ_k computed by our method for the `boeing('S')` example.

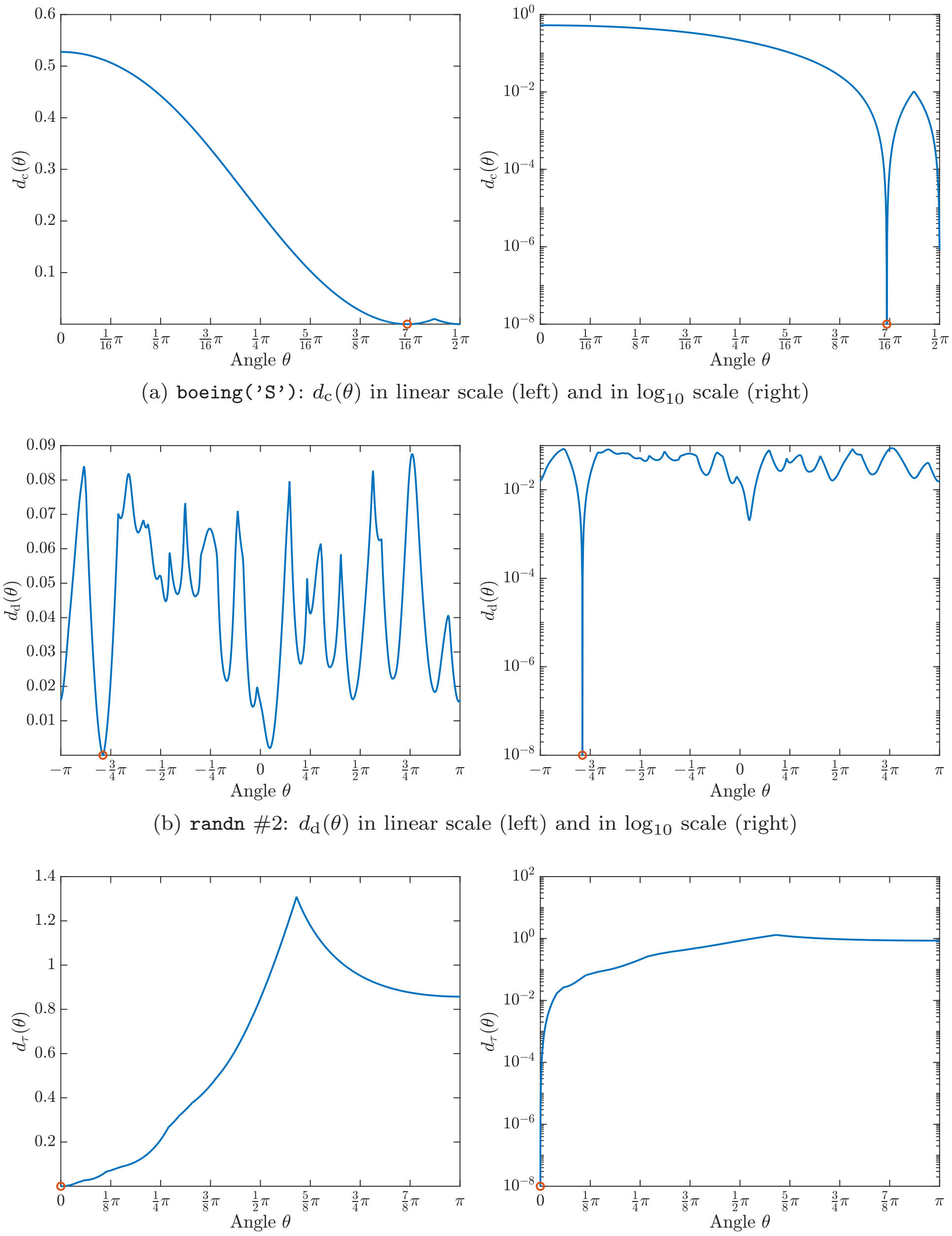
We now turn to comparing our new method for computing discrete-time Kreiss constants

	# of θ's evaluated per the kth restart and $\frac{\gamma_k - \gamma_{k+1}}{\gamma_k}$							
Problem	Restart 1		Restart 2		Restart 3		Restart 4	
`companion` (stab.)	15	`1e-02`	390	`1e-10`	**389**	—	—	—
`boeing('S')`	15	`7e-01`	15	`7e-01`	**32918**	—	—	—
`orrsommerfeld`	15	`9e-01`	**2598**	—	—	—	—	—
`convdiff` (mod.)	15	`3e-01`	15	`4e-02`	31	`3e-02`	**274**	—
`randn` #1 (stab.)	63	`1e-01`	**12324**	—	—	—	—	—
`randn` #2 (stab.)	15	`2e-01`	15	`2e-01`	127	`1e-01`	**18518**	—
`kahan` ($m = 20$)	15	`7e-01`	**177**	`3e-15`	—	—	—	—
`kahan` ($m = 30$)	15	`6e-01`	15	`2e-01`	**7137**	—	—	—

Table 2: For each restart using our new interpolation-based globality certificates, the left number is the total number of points at which Chebfun evaluated $d_{\mathrm{c}}(\theta)$, $d_{\mathrm{d}}(\theta)$, or $d_{\tau}(\theta)$ for the current estimate γ_k until either new starting points were found (which immediately restarts optimization) or Chebfun terminated on its own; bold font indicates the last certificate computed. The right number is the relative difference obtained by the next round of optimization to lower γ_k. Note that for `kahan` ($m = 20$), the last test produced new starting points but optimization was unable to meaningfully lower estimate γ_k further and so our code terminated after a round of optimization instead of after a certificate test.

with our earlier method of [Mit19, Section 5], again using three differently-sized test problems; see the middle three rows of Table 1. The first, `convdiff` (mod.), is the modified EigTool example matrix we used for Figure 2. For larger problems, we generated two complex-valued non-Hermitian matrices using `randn` and ensured that they were stable by scaling them so the spectral radius of each was 0.999; these examples are respectively called `randn` #1 (stab.) and `randn` #2 (stab.). Unlike all the other examples considered so far, the level sets of these two are not symmetric and so the full $[-\pi, \pi]$ domain of $d_{\mathrm{d}}(\theta)$ must be considered. All three matrices have very low Kreiss constants but are useful for demonstration since (1.7a) has multiple different local minima for each of them. Both methods were initialized at the same starting points, again chosen so at least one restart would be necessary for each problem. For the smallest example, `convdiff` (mod.), the estimates of $\mathcal{K}(A)$ computed by our new and earlier method agreed to machine precision (the relative difference was 9.4×10^{-16}), and we again see that our new method was faster even for $n = 10$. For the larger examples, it was again impractical to run the full algorithm of [Mit19, Section 5]; instead, for each of these two problems, we only recorded the time needed by `polyeig` in MATLAB to compute all the eigenvalues of a single instance of the $4n^2 \times 4n^2$ quadratic eigenvalue problem [Mit19, Equation (5.23)], using the final value of γ_k computed by our new method. This ended up only being possible for the `randn` #1 (stab.) example, which has dimension $n = 50$, and this single call to `polyeig` took 24.8 times longer than the time to run our new method in entirety. For `randn` #2 (stab.), which has dimension $n = 100$, `polyeig` terminated almost immediately due to running out of memory (on a computer with 192 GB of RAM); this underscores that our new method is also much less memory intensive. In Table 2, the number of points at which Chebfun evaluated $d_{\mathrm{d}}(\theta)$ for each certificate computation is shown; again relatively few values of θ are generally evaluated before a global minimizer is found in the optimization phases. The number of points evaluated to build the final interpolants for these examples again does not seem to be strongly associated with the matrix dimension. In Figure 3b, we show $d_{\mathrm{d}}(\theta)$ for the final value of γ_k computed by our method for the `randn #2 (stab.)` example.

Finally, we compare our new interpolation-based certificates for computing the distance to uncontrollability with the divide-and-conquer-based certificates developed in [GMO+06]. Recall that the divide-and-conquer technique has $\mathcal{O}(n^4)$ work on average and $\mathcal{O}(n^5)$ in the worst case, whereas computing our older Kreiss-constant certificates have $\mathcal{O}(n^6)$ work. Hence, for computing



(a) `boeing('S')`: $d_c(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right)

(b) `randn` #2: $d_d(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right)

(c) `kahan` ($m = 30$): $d_\tau(\theta)$ in linear scale (left) and in $\log_{10}$ scale (right)

Figure 3: The topmost subfigure shows $d_c(\theta)$ at the final value of γ_k computed by our new method for the `boeing('S')` example, in linear scale (left) and $\log_{10}$ scale (right). The circle denotes the angle of the best minimizer obtained by optimization and corresponds to the single place where $d_c(\theta) = 0$ (which is more easily seen in the $\log_{10}$ plot), confirming that γ_k is the globally minimal value. The same is done for $d_d(\theta)$ and `randn` #2 in the middle subfigure and for $d_\tau(\theta)$ and `kahan` ($m = 30$) in the bottom subfigure.

$\tau(A,B)$, it is not immediately clear whether our interpolation-based certificates will be as competitive, as they have a lower bound work complexity of $\Omega(n^3)$ and it is not clear whether there even is an upper bound. To assess this, we compared our own code to the `dist_uncont_hybrid` routine[4], which uses BFGS for optimization and divide-and-conquer-based certificates when its options are set as follows: `opts.method=1` and `opts.eig_method=1`. Though there are differences in the optimization implementations and setup, this is not a big issue as the computation time is by far dominated by computing the certificates. In [GMO+06, Section 4.3], real-valued examples were generated by setting A to different sizes of the `kahan` demo from EigTool and $B =$ `randn(`n`,`m`)`. We do the same here but for larger sizes; see the last two rows of Table 1 for the dimensions. For each problem, the two methods were initialized at the origin. The relative differences between the estimates computed by both methods were respectively 5.3×10^{-13} and 2.8×10^{-13} for `kahan` ($m = 20$) and `kahan` ($m = 30$), with our new method returning the slightly better (lower) estimates for both. Furthermore, our new method was respectively 99.8 and 37.9 times faster than running `dist_uncont_hybrid` on these two problems. The number of points evaluated for each restart is given in the last two rows of Table 2, while $d_\tau(\theta)$ is shown in Figure 3c for `kahan` ($m = 30$) using the final value of γ_k. Part of the reason why our new method was so fast on the smaller `kahan` ($m = 20$) example is because it did not actually complete computing the approximation to $d_\tau(\theta)$ on its last certificate test. Instead, Chebfun was halted quite early, as new starting points were found, but optimization could not significantly lower the objective function from these points so our method terminated.

Remark 6.1. *For all but one of our test problems, Chebfun satisfied its convergence criteria in building chebfuns of the final distance functions. However, for* `boeing('S')`*, Chebfun warned that it did not sufficiently resolve the last $d_\mathrm{c}(\theta)$, which is shown in Figure 3a. When this happens, technically globality is not asserted by the interpolant approximation, but it also generally means that the distance function is evaluated at many different values of θ (specifically 32918 for this example) without any new starting points being discovered. Hence it seems likely that the computed estimate γ_k is nevertheless globally optimal, and of course there are no new starting points to be found if γ_k is indeed globally optimal.*

6.2 Additional acceleration via parallel processing

Both the optimization phases and our new interpolation-based certificates have embarrassingly parallel components. Optimization can be run in parallel whenever there are multiple starting points, which would hopefully increase the chance of finding a global minimizer on any given iteration, thus reducing the number of the more costly certificate computations. For the interpolation-based certificates, any time Chebfun provides a vector of multiple values of θ, obtaining the corresponding function values of $d_\mathrm{c}(\theta)$, $d_\mathrm{d}(\theta)$, or $d_\tau(\theta)$ is an embarrassingly parallel task. For brevity here, we only consider accelerating this portion of our new method since it is the dominant cost, which was done by simply evaluating each vector of θ values using a `parfor` loop, with `parpool(cores)` called for `cores` set to 2, 4, 8, 16, and 32. We tested this in two configurations: with the Chebfun preference `'min_samples'` kept at its default value of 17 and then again with it increased to 65.

In Table 3, we show the resulting speedups compared to our single-core configuration used in §6.1 for the three largest of our test examples. With `'min_samples'` set to its default of 17, the best speedups achieved ranged from 6.0 to 9.0. While this may seem a rather low utilization of a 32-core machine, this is because the average number of θ values provided at a time by Chebfun was also rather low, i.e., 15.8 to 22.2. In other words, in this configuration, it is not possible to achieve speedups larger than 15.8 to 22.2. Though not shown in Table 3, on our medium-sized problems, parallel processing resulted in a speedup of roughly two times, while on the two $n = 10$ examples, parallel processing generally increased the running times. The low average size of the vectors of θ values provided by Chebfun means that the overhead of entering and exiting the

[4] Available at `http://home.ku.edu.tr/~emengi/software/robuststability.html`

	Speedup per # of cores					Vector of θ's	
Problem	2	4	8	16	32	#	Avg. Size
Chebfun `min_samples`: 17							
`orrsommerfeld`	2.6	3.8	5.2	6.0	5.4	157	16.5
`randn` #2 (stab.)	3.0	4.6	6.4	8.1	9.0	780	22.2
`kahan` ($m = 30$)	2.7	4.1	5.6	7.2	8.3	453	15.8
Chebfun `min_samples`: 65							
`orrsommerfeld`	2.7	4.0	5.5	6.3	5.5	138	20.2
`randn` #2 (stab.)	3.0	5.2	7.1	9.0	9.6	592	30.0
`kahan` ($m = 30$)	2.7	4.3	6.1	8.1	9.4	402	19.1

Table 3: The speedups for the number of θ's evaluated per second while Chebfun is building the final interpolant for the three largest problems; speedups are done with respect to the rate, instead of the total time to build each interpolant, since the total number of points Chebfun evaluated was not always the same as the single-core configuration used in §6.1. The last two columns, "#" and "Avg. Size", respectively show the number of times Chebfun requested a vector of different values of θ to be evaluated and the average length of these vectors. These average lengths give upper bounds on the best possible speedups, while the pair of values together show that there is likely high overhead due to entering and existing the `parfor` loop many times in order for Chebfun to evaluate more and more points.

`parfor` loop is incurred many times, which greatly reduces how much speedup can be attained, particularly when n is rather small. With `'min_samples'` set to 65, the average number of θ values provided at a time increased to 19.1 to 30.0, but the corresponding best speedups, now 6.3 to 9.6 times faster, only modestly improved.

We analyzed the Chebfun code to ascertain how else the average vector size might be increased and the total number of vectors provided decreased. Perhaps the biggest influence is the `findJump` routine inside `@fun/detectEdge.m`, which does bisection to detect singularities and thus requests only a single function value per iteration, for many iterations. We modified `findJump` to instead do k-sectioning for integers $k > 2$ and found that our new version dramatically increased the overall average vector length if k was sufficiently large, particularly since it also dramatically reduced the number of iterations `findJump` needed. Another cause is related to the fact that Chebfun often approximates functions, particularly nonsmooth ones, not by a single polynomial interpolant but a concatenation of them. For each piece, a final safety test for accuracy (`@chebtech/sampleTest.m`) is done using a pair of *hard-coded* points in the interval $[-1, 1]$; internally, each piece has a domain of $[-1, 1]$, which is rescaled to the region that it is approximating. With parallel processing, it would be more efficient to speculatively evaluate these two fixed values for each piece, by batching them in with the piece's vector of initial sample points, and store this pair of function values for recall later.

Remark 6.2. *Parallel eigensolvers such as [BKS14] could also be used to accelerate solving the large eigenvalue problems in the older certificate tests of [Mit19] and [Gu00], but this would not reduce their high memory requirements nor does it seem likely that it would be competitive with our interpolation-based certificates even using serial computation, let alone parallel computation.*

7 Concluding remarks

We have seen that our new interpolation-based globality certificates are substantially more efficient than the existing state-of-the-art certificates of [Mit19] for Kreiss constants and those of [Gu00, GMO$^+$06] for the distance to uncontrollability. Although our new approach assumes the relevant one-variable distance functions will be adequately sampled to find their zeros (if any), this seems a rather mild assumption in practice, as they will be zero on *positive measure* subsets of their domains before a global minimizer has been obtained.

One thing we have not investigated here is whether our new assumptions: zero is not an eigenvalue of A for continuous-time $\mathcal{K}(A)$, γ^2 is not an eigenvalue of AA^* for discrete-time $\mathcal{K}(A)$, and γ^2 is not eigenvalue of AA^*+BB^* for $\tau(A,B)$, are at all restrictive in practice. In the presence of rounding errors, it is unlikely that these assumptions would actually be violated numerically. Furthermore, just discarding any exactly zero eigenvalues of the relevant matrix pencils when computing the distance functions might be okay in practice; although doing so could introduce discontinuities, Chebfun is rather capable of approximating functions with discontinuities.

Finally, we note that our new interpolation-based certificates, which sweep the complex plane with rays from the origin to locate level sets, could be used to solve other global optimization problems of singular value functions of two real variables.

Acknowledgments

The author is very grateful to Michael L. Overton for supporting several research visits to the Courant Institute in New York, where the main idea for this paper was conceived, and for many helpful comments while preparing this manuscript. The author is also grateful to Peter Benner for pointing out [LY91] and informative discussions on the possibility of creating structure-preserving eigensolvers for the indefinite generalized Hermitian eigenvalue problems that appear here.

A Real symmetry conditions for the level sets of $\tau(A, B)$

Lemma A.1. *Let $A \in \mathbb{C}^{n \times n}$ with $A = A^*$, $B \in \mathbb{C}^{n \times m}$, and $\sigma_k(M)$ denote the kth singular value of a matrix M. Then*

$$\sigma_k\left(\begin{bmatrix} A - \lambda I & B \end{bmatrix}\right) = \sigma_k\left(\begin{bmatrix} A - \overline{\lambda} I & B \end{bmatrix}\right)$$

if either (i) *A and B are both real-valued matrices or* (ii) *A is Hermitian.*

Proof. Case (i) holds since conjugation does not change singular values, i.e.,

$$\sigma_k\left(\begin{bmatrix} A - \lambda I & B \end{bmatrix}\right) = \sigma_k\left(\begin{bmatrix} \overline{A} - \overline{\lambda} I & \overline{B} \end{bmatrix}\right) = \sigma_k\left(\begin{bmatrix} A - \overline{\lambda} I & B \end{bmatrix}\right),$$

where the middle equivalence uses the fact that $A = \overline{A}$ and $B = \overline{B}$ since both are real. Case (ii) follows from the equivalence $\sigma_k(M) \iff \sigma_k^2 \in \Lambda(MM^*)$:

$$\begin{aligned}
\sigma_k\left(\begin{bmatrix} A - \lambda I & B \end{bmatrix}\right) \quad &\iff \quad \sigma_k^2 \in \Lambda\left(\begin{bmatrix} A - \lambda I & B \end{bmatrix} \begin{bmatrix} A^* - \overline{\lambda} I \\ B^* \end{bmatrix}\right) \\
&\iff \quad \sigma_k^2 \in \Lambda\left(\begin{bmatrix} AA^* - \lambda A^* - \overline{\lambda} A + |\lambda|^2 I + BB^* \end{bmatrix}\right) \\
&\iff \quad \sigma_k^2 \in \Lambda\left(\begin{bmatrix} AA^* - \lambda A - \overline{\lambda} A^* + |\lambda|^2 I + BB^* \end{bmatrix}\right) \\
&\iff \quad \sigma_k^2 \in \Lambda\left(\begin{bmatrix} A - \overline{\lambda} I & B \end{bmatrix} \begin{bmatrix} A^* - \lambda I \\ B^* \end{bmatrix}\right) \\
&\iff \quad \sigma_k\left(\begin{bmatrix} A - \overline{\lambda} I & B \end{bmatrix}\right).
\end{aligned}$$

where the third line uses the assumption that $A = A^*$. □

References

[BKS14] P. Benner, M. Köhler, and J. Saak. Fast approximate solution of the non-symmetric generalized eigenvalue problem on multicore architectures. In M. Bader, A. Bode-and, H.-J. Bungartz, M. Gerndt, G. R. Joubert, and F. Peters, editors, *Parallel*

Computing: Accelerating Computational Science and Engineering (CSE), volume 25 of *Advances in Parallel Computing*, pages 143–152. IOS Press, 2014.

[BLO03] 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.

[BLO04] J. V. Burke, A. S. Lewis, and M. L. Overton. Pseudospectral components and the distance to uncontrollability. *SIAM J. Matrix Anal. Appl.*, 26(2):350–361, 2004.

[BM17] P. Benner and T. Mitchell. Extended and improved criss-cross algorithms for computing the spectral value set abscissa and radius. e-print arXiv:1712.10067, arXiv, December 2017. math.OC.

[Bye88] R. Byers. A bisection method for measuring the distance of a stable to unstable matrices. *SIAM J. Sci. Statist. Comput.*, 9:875–881, 1988.

[Bye90] R. Byers. Detecting nearly uncontrollable pairs. In M. A. Kaashoek, J. H. Schuppen, and A. C. M. Ran, editors, *Signal processing, scattering and operator theory, and numerical methods*, volume 3 of *Proceedings of the International Symposium MTNS-89, Amsterdam 1989*, pages 447–457. Birkhäuser, Boston, MA, 1990.

[Eis84] R. Eising. Between controllable and uncontrollable. *Syst. Cont. Lett.*, 4(5):263–264, 1984.

[EK17] M. Embree and B. Keeler. Pseudospectra of matrix pencils for transient analysis of differential-algebraic equations. *SIAM J. Matrix Anal. Appl.*, 38(3):1028–1054, 2017.

[GMO$^+$06] M. Gu, E. Mengi, M. L. Overton, J. Xia, and J. Zhu. Fast methods for estimating the distance to uncontrollability. *SIAM J. Matrix Anal. Appl.*, 28(2):477–502, 2006.

[GN93] M. Gao and M. Neumann. A global minimum search algorithm for estimating the distance to uncontrollability. *Linear Algebra Appl.*, 188/189:305–350, 1993.

[Gu00] M. Gu. New methods for estimating the distance to uncontrollability. *SIAM J. Matrix Anal. Appl.*, 21(3):989–1003, 2000.

[Kre62] H.-O. Kreiss. Über die Stabilitätsdefinition für Differenzengleichungen die partielle Differentialgleichungen approximieren. *BIT Numerical Mathematics*, 2(3):153–181, 1962.

[LY91] P. Lancaster and Q. Ye. Variational and numerical methods for symmetric matrix pencils. *Bull. Austral. Math. Soc.*, 43(1):1–17, 1991.

[Meh04] C. Mehl. Jacobi-like algorithms for the indefinite generalized Hermitian eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 25(4):964–985, 2004.

[Men06] E. Mengi. *Measures for Robust Stability and Controllability.* PhD thesis, New York University, New York, NY 10003, USA, September 2006. `https://cs.nyu.edu/media/publications/mengi_emre.pdf`.

[Mit] T. Mitchell. ROSTAPACK: RObust STAbility PACKage. `http://timmitchell.com/software/ROSTAPACK`.

[Mit19] T. Mitchell. Computing the Kreiss constant of a matrix. e-print arXiv:1907.06537, arXiv, July 2019. math.OC.

[TE05] L. N. Trefethen and M. Embree. *Spectra and pseudospectra: The behavior of nonnormal matrices and operators.* Princeton University Press, Princeton, NJ, 2005.

[Wri02] T. G. Wright. EigTool. `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`, 2002.