SIAM J. MATRIX ANAL. APPL.
Vol. 42, No. 2, pp. 578–607

FAST INTERPOLATION-BASED GLOBALITY CERTIFICATES FOR COMPUTING KREISS CONSTANTS AND THE DISTANCE TO UNCONTROLLABILITY*

TIM MITCHELL†

Abstract. We propose a new approach to computing global minimizers of singular value functions in two real variables. Specifically, we present new algorithms to compute the Kreiss constant of a matrix and the distance to uncontrollability of a linear control system, both to arbitrary accuracy. Previous state-of-the-art methods for these two quantities rely on 2D level-set tests that are based on solving large eigenvalue problems. Consequently, these methods are costly, i.e., $\mathcal{O}(n^6)$ work using dense eigensolvers, and often multiple tests are needed before convergence. Divide-and-conquer techniques have been proposed that reduce the work complexity to $\mathcal{O}(n^4)$ on average and $\mathcal{O}(n^5)$ in the worst case, but these variants are nevertheless still very expensive and can be numerically unreliable. In contrast, our new interpolation-based globality certificates perform level-set tests by building interpolant approximations to certain one-variable continuous functions that are both relatively cheap and numerically robust to evaluate. Our new approach has an $\mathcal{O}(kn^3)$ work complexity and uses $\mathcal{O}(n^2)$ memory, where k is the number of function evaluations necessary to build the interpolants. Not only is this interpolation process mostly "embarrassingly parallel," but also low-fidelity approximations typically suffice for all but the final interpolant, which must be built to high accuracy. Even without taking advantage of the aforementioned parallelism, k is sufficiently small that our new approach is generally orders of magnitude faster than the previous state of the art.

Key words. transient growth, robust stability, controllability, pseudospectra

AMS subject classifications. 15A16, 37C75, 39A22, 39A30, 65F30, 65F60

DOI. 10.1137/20M1358955

Notation. $\|\cdot\|$ denotes the spectral norm, $\sigma_{\min}(\cdot)$ the smallest singular value, $\Lambda(\cdot)$ the spectrum, and $\kappa(\cdot)$ the condition number of a matrix with respect to the spectral norm. A matrix pencil $A - \lambda B$ and its spectrum are denoted by (A, B) and $\Lambda(A, B)$, respectively, and (A, B) is a regular matrix pencil if there exists at least one $\lambda \in \mathbb{C}$ such that $\det(A - \lambda B) \neq 0$. A matrix $A \in \mathbb{C}^{2n\times 2n}$ is (skew-)Hamiltonian if $(JA)^* = JA$ ($A^*J = JA$), where $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$. A matrix pencil (A, B) with $A, B \in \mathbb{C}^{2n\times 2n}$ is skew-Hamiltonian-Hamiltonian (sHH) if B is skew-Hamiltonian and A is Hamiltonian. Euler's number, $2.71828\ldots$, is denoted by e, while $\operatorname{bd}\mathcal{A}$ and $\operatorname{int}\mathcal{A}$ are the boundary and interior of a set $\mathcal{A}$, respectively.

1. Introduction.

We begin with two important quantities that can be written as global optimization problems of certain singular value functions in two real variables and describe existing algorithms for computing these quantities and their limitations. In this discussion, we also establish necessary background and context for understanding our new approach.

1.1. The distance to uncontrollability.

Consider the linear control system

$$\dot{x} = Ax + Bu, \tag{1.1}$$

*Received by the editors October 3, 2019; accepted for publication (in revised form) by M. A. Freitag January 11, 2021; published electronically April 8, 2021.
https://doi.org/10.1137/20M1358955
Funding: The author's visits to the Courant Institute of Mathematical Sciences, New York University, were supported by National Science Foundation grant DMS-1620083.
†Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106 Germany (mitchell@mpi-magdeburg.mpg.de).

where $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and the state $x \in \mathbb{C}^n$ and control input $u \in \mathbb{C}^m$ are dependent on time. System (1.1) is *controllable* if, for any pair of initial and final states $x(0)$ and $x(T)$, $T > 0$, there exists a control function $u(\cdot)$ that takes $x(0)$ to $x(T)$. However, a more robust measure is the *distance to uncontrollability*, which was introduced in [23] and is denoted here by $\tau(A, B)$. Given a system (1.1), $\tau(A, B)$ specifies the distance to the nearest matrix pair $A_{\text{uc}}, B_{\text{uc}}$ such that $\dot{x} = A_{\text{uc}}x + B_{\text{uc}}u$ is uncontrollable, with the distance being zero if (1.1) is uncontrollable and positive otherwise.[1] In [10], Eising showed that $\tau(A, B)$ is equal to the globally minimal value of the following singular value function:

$$\tau(A, B) = \min_{z\in\mathbb{C}} \sigma_{\min}\left(\begin{bmatrix} A - zI & B \end{bmatrix}\right). \tag{1.2}$$

While many methods have been proposed for $\tau(A, B)$ over the years (see [14, p. 990] for a historical overview), the first polynomial-time algorithm to correctly estimate $\tau(A, B)$ to within a constant factor is due to Gu [14]. The basis of Gu's algorithm is a 2D level-set test. Given a guess $\gamma \geq 0$ for the value of $\tau(A, B)$ and a parameter $\eta > 0$, Gu's 2D level-set test verifies whether or not there exists one or more points $\tilde{z} \in \mathbb{C}$ such that

$$\gamma = \sigma_{\min}\left(\begin{bmatrix} A - \tilde{z}I & B \end{bmatrix}\right) = \sigma_{\min}\left(\begin{bmatrix} A - (\tilde{z} + \eta)I & B \end{bmatrix}\right) \tag{1.3}$$

holds. Moreover, if such points exist, Gu's test also returns their values. Clearly $\tau(A, B) \leq \gamma$ holds if (1.3) is satisfied by some $\tilde{z}$. However, if there are no such points, [14, Theorem 3.1] states that the following lower bound must instead hold:

$$\tau(A, B) > \gamma - \tfrac{\eta}{2}. \tag{1.4}$$

By starting with $\gamma_0 \geq \tau(A, B)$ and using $\gamma_{k+1} \coloneqq \frac{1}{2}\gamma_k$ and $\eta_{k+1} \coloneqq \gamma_{k+1}$, Gu's method for $\tau(A, B)$ repeatedly applies his 2D level-set test until it no longer finds any points $\tilde{z} \in \mathbb{C}$ satisfying (1.3). Assuming exact arithmetic is used, at termination, the last estimate is guaranteed to be within a factor of two of $\tau(A, B)$ (in either direction). The cost of Gu's method is dominated by the need to compute all real eigenvalues of a large *generalized* eigenvalue problem of order $2n^2$ for each 2D level-set test, as described in [14, section 3.2]. Consequently, when using standard dense eigensolvers, such as those based on the QZ algorithm, Gu's estimation algorithm has an $\mathcal{O}(n^6)$ work complexity[2] and requires $\mathcal{O}(n^4)$ memory. In practice, this limits Gu's method to all but the smallest of problems, and furthermore, in inexact arithmetic, his 2D level-set test can fail due to rounding errors, particularly as η gets closer and closer to zero; see [6, p. 358].

Using Gu's 2D level-set test, Burke, Lewis, and Overton [6] proposed the first two algorithms to compute $\tau(A, B)$ to arbitrary accuracy, again assuming exact arithmetic. Their first method [6, Algorithm 5.2] uses Gu's test in a trisection iteration in an effort to minimize the speed at which $\eta \to 0$ as trisection converges to $\tau(A, B)$. In turn, this helps reduce the chances of Gu's test failing numerically before the estimate to $\tau(A, B)$ has been sufficiently resolved. We forgo describing trisection in detail, but mention that trisection is not a panacea, since if $\tau(A, B)$ is very small, so must η be

[1] A new alternative for assessing the distance to uncontrollability of systems with input *and output* was recently proposed in [18, 12], but here we focus on the standard definition.

[2] As in [6], 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.

to resolve $\tau(A,B)$ to even a single digit of accuracy; see [22, Lemma B.1 and Corollary B.2]. The authors' second method [6, Algorithm 5.3] combines Gu's test and local optimization to yield an optimization-with-restarts iteration. As $\sigma_{\min}\left(\left[\begin{smallmatrix} A-zI & B \end{smallmatrix}\right]\right)$ is Lipschitz and will generally be smooth, even at minimizers, finding locally optimal approximations to $\tau(A,B)$ via standard optimization techniques is straightforward and can be done with few function evaluations. Computing the value, gradient, and Hessian of $\sigma_{\min}\left(\left[\begin{smallmatrix} A-zI & B \end{smallmatrix}\right]\right)$ is $\mathcal{O}(n^3)$ work via standard dense SVD methods, while the function value and gradient can be obtained in just $\mathcal{O}(n)$ work via sparse SVD methods. Thus, given a local minimizer z_k of (1.2) with $f_k = \sigma_{\min}\left(\left[\begin{smallmatrix} A-z_kI & B \end{smallmatrix}\right]\right)$, [6, Algorithm 5.3] uses Gu's test with carefully chosen values for parameters γ and η such that the algorithm terminates if f_k is sufficiently close to $\tau(A,B)$ in a relative sense. If not, Gu's test still provides new level-set points such that running optimization from them must yield a better (lower) minimizer. Since the objective function in (1.2) is semialgebraic, it has a finite number of locally minimal function values; see [6, p. 359]. As a result, [6, Algorithm 5.3] must terminate at a globally optimal minimizer within a finite number of optimization restarts (for brevity, throughout the paper we assume that optimization finds stationary points exactly). In practice, the number of restarts is typically only a handful, which makes it many times faster than the trisection iteration.

Shortly thereafter, [15] showed how the large generalized eigenvalue problem in Gu's 2D level-set test can be reduced to a *standard* eigenvalue problem (but still of order $2n^2$), and then proposed a divide-and-conquer technique to compute the relevant eigenvalues in $\mathcal{O}(n^4)$ work on average and $\mathcal{O}(n^5)$ in the worst case. While divide-and-conquer enables asymptotically faster versions of all of the methods of [14, 6] described above, it does not address the aforementioned numerical issues inherent in Gu's 2D level-set test. In fact, divide-and-conquer introduces additional numerical uncertainties, as it relies on sparse shift-and-invert eigensolver techniques. As we mentioned in [22, section 8], one issue is that divide-and-conquer assumes that such eigensolvers always return the closest eigenvalues to a given shift, which, while reasonable, is not always true in practice. Furthermore, sparse eigensolvers such as `eigs` in MATLAB can have convergence issues when the norm of the matrix in question gets large; see [15, p. 500]. Nevertheless, it is not always clear whether divide-and-conquer will be more or less reliable than Gu's original test using dense eigensolvers, and indeed the experiments of [15, section 4] do demonstrate that divide-and-conquer can be a much faster and effective alternative for computing $\tau(A,B)$. For a thorough discussion of the numerical difficulties of both Gu's original approach and divide-and-conquer, see [15, sections 4.1, 5.2, and 5.3] and the references within.

Finally, we recently showed how the numerical reliability of all of these $\tau(A,B)$ methods can be greatly improved via a crucial reinterpretation and modified version of Gu's 2D level-set test; see [22, Key Remark 6.3].

1.2. The Kreiss constant of a matrix. We now turn to another important quantity, namely, the *Kreiss constant* of a matrix $A \in \mathbb{C}^{n\times n}$, which comes in continuous- and discrete-time variants that respectively bound the transient behavior of $\dot{x} = Ax$ and $x_{k+1} = Ax_k$. More specifically, the discrete-time version of the Kreiss Matrix Theorem [17], after being refined by many authors over nearly thirty years, states that [24, Theorem 18.1]

$$\mathcal{K}(A) \le \sup_{k\ge 0} \|A^k\| \le en\mathcal{K}(A), \tag{1.5}$$

where the *Kreiss constant* $\mathcal{K}(A)$ has two equivalent definitions [24, p. 143],

$$\mathcal{K}(A) = \sup_{z\in\mathbb{C},|z|>1} (|z|-1)\|(zI-A)^{-1}\|, \tag{1.6a}$$

$$= \sup_{\varepsilon>0} \frac{\rho_\varepsilon(A)-1}{\varepsilon}, \tag{1.6b}$$

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

$$\rho_\varepsilon(A) = \max\{|z| : z \in \Lambda(A+\Delta), \|\Delta\| \le \varepsilon\} \tag{1.7a}$$

$$= \max\{|z| : z \in \mathbb{C}, \|(zI-A)^{-1}\| \ge \varepsilon^{-1}\}. \tag{1.7b}$$

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 $\rho(A) \le 1$, then $\mathcal{K}(A) = 1$, which is the minimum value $\mathcal{K}(A)$ can take.

The continuous-time Kreiss Matrix Theorem states that [24, Theorem 18.5]

$$\mathcal{K}(A) \le \sup_{t\ge0} \|\mathrm{e}^{tA}\| \le \mathrm{e}n\mathcal{K}(A), \tag{1.8}$$

where this version of $\mathcal{K}(A)$ also has two equivalent definitions [24, eqn. (14.7)],

$$\mathcal{K}(A) = \sup_{z\in\mathbb{C},\mathrm{Re}\, z>0} (\mathrm{Re}\, z)\|(zI-A)^{-1}\|, \tag{1.9a}$$

$$= \sup_{\varepsilon>0} \frac{\alpha_\varepsilon(A)}{\varepsilon}, \tag{1.9b}$$

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

$$\alpha_\varepsilon(A) = \max\{\mathrm{Re}\, z : z \in \Lambda(A+\Delta), \|\Delta\| \le \varepsilon\} \tag{1.10a}$$

$$= \max\{\mathrm{Re}\, z : z \in \mathbb{C}, \|(zI-A)^{-1}\| \ge \varepsilon^{-1}\}. \tag{1.10b}$$

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, $\mathcal{K}(A) \ge 1$ always holds and $\mathcal{K}(A) = 1$ if A is normal and $\alpha(A) \le 0$.

In [22], we introduced the first globally convergent algorithms to compute both continuous- and discrete-time Kreiss constants to arbitrary accuracy. Prior to this, it was only possible to estimate $\mathcal{K}(A)$ using supervised techniques, i.e., where a user is an active participant of the process. In [19, Chapter 3.4.1] and [11], Kreiss constants were approximated by plotting (1.6b) or (1.9b) and simply taking the maximum of the resulting curve. Meanwhile, Kreiss constant estimation via plotting $\|\mathrm{e}^{tA}\|$ with respect to t or $\|A^k\|$ with respect to k or by finding local maximizers of (1.6b) or (1.9b) via optimization is discussed in [24, Chapters 14 and 15]. Plotting and/or grid 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. In contrast, under sufficient regularity conditions, optimization techniques have high fidelity in finding local maximizers, often with relatively few function evaluations. However, as the optimization problems in (1.6) and (1.9) are typically nonconvex, general optimization solvers cannot guarantee that a global maximizer is found, and estimates from local minimizers can be arbitrarily bad approximations to $\mathcal{K}(A)$. Even if one happens to know a relatively small bounded region containing a global maximizer of the optimization problems in (1.6) and (1.9),

to guarantee any level of accuracy, this region must still be sufficiently sampled (for plotting or grid techniques) or contain no other stationary points (for optimization). Knowing such a region *and* how much sampling is required or that it contains no other stationary points is not typical, at least not without user experimentation. Moreover, if transient behavior occurs on a fast time scale, such regions can be very small and thus hard to find, particularly without fine-grained sampling. As noted by Mengi [19, section 6.2.2], "in general it is difficult to guess *a priori* which ε value is most relevant for the transient peak [of (1.6b) or (1.9b)]."

For our recent algorithms to compute Kreiss constants with theoretical guarantees [22], we actually worked with the inverses of (1.6a) and (1.9a), respectively:

$$\mathcal{K}(A)^{-1} = \inf_{|z|>1} \sigma_{\min}\left(\frac{zI-A}{|z|-1}\right) \qquad \text{(discrete-time)}, \tag{1.11a}$$

$$\mathcal{K}(A)^{-1} = \inf_{\operatorname{Re} z>0} \sigma_{\min}\left(\frac{zI-A}{\operatorname{Re} z}\right) \qquad \text{(continuous-time)}. \tag{1.11b}$$

In this form, it is easier to see that the optimization problems in (1.11) have some similarity to (1.2), which naturally leads to the question of whether or not any of the aforementioned $\tau(A,B)$ methods could be adapted to computing $\mathcal{K}(A)$. Like the $\tau(A,B)$ setting, the objective functions in (1.11) are semialgebraic, so they have a finite number of locally minimal values; hence, properly designed optimization-with-restart algorithms will converge to $\mathcal{K}(A)^{-1}$ within a finite number of restarts. Optimization can also robustly and efficiently find (feasible) minimizers of (1.11) in order to obtain locally optimal approximations to $\mathcal{K}(A)^{-1}$, even for large scale problems. For complete details on both of these points, see our previous comments in subsection 1.1 and [22, section 3]. However, as shown in [22, section 4], there are in fact fundamental differences between computing the distance to uncontrollability, and Kreiss constants and existing $\tau(A,B)$ methods do not extend directly. Nevertheless, for the objective functions in (1.11), we developed several different Kreiss constant analogues of Gu's [14, Theorem 3.1], which, along with several new 2D level-set tests, enable three different $\mathcal{K}(A)$ iterations [22].

When computing continuous-time $\mathcal{K}(A)$, the first of these is based on a new 2D level-set test that, similar to Gu's $\tau(A,B)$ test for (1.3), looks for pairs of level-set points of the objective function in (1.11b) that are a *fixed-distance* η apart. However, the aforementioned $\tau(A,B)$ trisection and optimization-with-restart algorithms of [6] cannot be used with this new test because a meaningful lower bound for $\mathcal{K}(A)^{-1}$, like (1.4), is *not* asserted when no such level-set pairs are detected; see [22, section 4]. But, by combining optimization-with-restarts with a *backtracking procedure*, it was possible to use this fixed-distance test to devise a globally convergent iteration for $\mathcal{K}(A)$; see [22, section 5]. This new level-set involves computing all real eigenvalues of a generalized eigenvalue problem $\mathcal{A}_1 - \lambda\mathcal{A}_2$ of order $4n^2$ [22, eqn. (5.5)], where $\mathcal{A}_2$ is singular with rank $2n^2$. As noted in [22, section 5.3], it does not seem possible to analytically reduce the order of $\mathcal{A}_1 - \lambda\mathcal{A}_2$ to $2n^2$ or to a standard eigenvalue problem via the techniques of [15] for Gu's $\tau(A,B)$ level-set test.

Meanwhile, in [22, section 6], we devised a second 2D level-set test for (1.11b) that looks for pairs of level-set points that are a certain *variable distance* apart involving η. This too leads to solving a generalized eigenvalue problem of order $4n^2$, $\mathcal{B}_1 - \lambda\mathcal{B}_2$ [22, eqn. (6.6)], but here $\mathcal{B}_2$ is nonsingular and we derived an explicit form for its inverse. Differences in numerical reliability between the fixed- and variable-distance tests are still not entirely clear, but one advantage of the variable-distance test is that it *does*

assert that the lower bound $\mathcal{K}(A)^{-1} > \gamma - \frac{\eta}{2}$ must hold whenever no such level-set pairs are detected. Thus, this variable-distance 2D level-set test enables two more $\mathcal{K}(A)$ iterations, which respectively use trisection and optimization-with-restarts *without backtracking*. In practice, optimization-with-restarts without backtracking generally needs the least number of level-set tests, while trisection needs far more than either of the optimization-based iterations.

To compute discrete-time $\mathcal{K}(A)$, we also developed three analogues of these iterations [22, section 7], which, at a very high level, work similarly. However, the underlying new 2D level-set tests for the objective function in (1.11a) are substantially different and more complicated and expensive. In particular, they require solving *quadratic* eigenvalue problems of order $4n^2$.

Like the $\tau(A, B)$ methods, these $\mathcal{K}(A)$ methods do $\mathcal{O}(n^6)$ work when using dense eigensolvers. However, solving the larger and generalized/quadratic eigenvalue problems is substantially more expensive and requires much more memory, particularly for discrete-time $\mathcal{K}(A)$. In [22, sections 5.4, 6.3, and 7.4], we also developed divide-and-conquer versions of all of these $\mathcal{K}(A)$ algorithms. While these variants do $\mathcal{O}(n^4)$ work on average and $\mathcal{O}(n^5)$ in the worst case and have dramatically lower memory requirements, we noted in [22, section 8] that divide-and-conquer for the $\mathcal{K}(A)$ setting does not appear to be very reliable in practice. All our 2D level-set tests for $\mathcal{K}(A)$ also use our improved technique that is explained in [22, Key Remark 6.3].

1.3. Motivation and contribution of the paper. As we have just seen, the state-of-the-art methods for computing $\tau(A, B)$ and $\mathcal{K}(A)$ are based on a 2D level-set test methodology that intrinsically involves solving very large eigenvalue problems. Even in their faster divide-and-conquer variants, these algorithms are prohibitively expensive. Moreover, the convergence guarantees of these methods assume exact computation, but rounding errors in computed eigenvalues may cause the methods to fail numerically. Our aforementioned modified technique to perform these 2D level-set tests more reliably, while effective and in fact crucial for robust codes, does not address all the numerical pitfalls of these methods.

In this paper, we address both these high-cost and reliability issues by proposing a new methodology for computing quantities whose values are given by global optimization of singular value functions in two real variables. We do this by developing new level-set tests for optimization-with-restarts-based methods, which we call *interpolation-based globality certificates* and that work by sufficiently resolving certain one-variable continuous functions over a *finite interval known a priori*. These new functions are reasonably well behaved and relatively cheap and robust to evaluate, all of which makes high-fidelity approximation via interpolation practical. Our new $\tau(A, B)$ and $\mathcal{K}(A)$ methods have $\mathcal{O}(kn^3)$ work complexity and require $\mathcal{O}(n^2)$ memory, where k, the total number of function evaluations incurred to build the interpolants, is such that our new methods are orders of magnitude faster than the previous state of the art. Moreover, additional significant speedups can be attained via parallel processing, since function evaluations for interpolation are "embarrassingly parallel." Our "strength in numbers" interpolation-based approach also has numerical benefits, as global convergence does not crucially hinge upon any *single* computation being susceptible to rounding error, which is not true for almost all of the 2D level-set test methods discussed in subsections 1.1 and 1.2 (the sole exception being the Kreiss constant iteration using backtracking). The trade-off we have made here is that instead of putting our faith in accurately computing eigenvalues of very large eigenvalue problems, we assume that approximation via interpolation is reliable enough to be used

as a subroutine. In this sense, our new approach can be considered complementary to our earlier efforts of [22], as they are built on very different foundations.

In section 2 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 derived, respectively, in sections 3 and 4. Numerical experiments are given in section 5, while concluding remarks are made in section 6.

2. A new approach for computing continuous-time Kreiss constants. We now propose a new optimization-with-restarts-based algorithm for computing continuous-time $\mathcal{K}(A)$, i.e., a new method to find a global minimizer of (1.11b). As previously mentioned, local minimizers of (1.11b) can be found relatively cheaply, and its objective function has a finite number of locally minimal values. Thus, given a corresponding level-set test, only a finite number of optimization restarts are necessary to compute $\mathcal{K}(A)^{-1}$, and equivalently $\mathcal{K}(A)$, to arbitrary accuracy. However, unlike the earlier methods discussed in the introduction, we abandon the concept of looking for *pairs of points on the γ-level set* of the given singular value function for which a global minimizer is sought. Instead, we focus on devising a new type of level-set test, which, given some $\gamma \geq \mathcal{K}(A)^{-1}$ corresponding to a minimizer of (1.11b), answers the question, Are there other points on this same level set, and if so, where are they?

For (1.11b), minimizers should be computed using Cartesian coordinates (see [22, section 3.1] for full details), but note that all of our *interpolation-based globality certificates* to detect level-set points are based on polar coordinates, even in continuous-time settings. Thus, consider (1.11b) parameterized in polar coordinates:

$$g(r,\theta) \coloneqq \sigma_{\min}(G(r,\theta)) \quad \text{and} \quad G(r,\theta) \coloneqq \frac{re^{\mathbf{i}\theta}I - A}{r\cos\theta}, \tag{2.1}$$

so

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

As computing $\mathcal{K}(A)$ is trivial if either A is normal or $\alpha(A) > 0$, we assume neither holds. For reasons that will become clear momentarily, we also assume that $0 \notin \Lambda(A)$.

2.1. Level sets of $g(r,\theta)$ and a 1D radial level-set test. For a fixed $\theta \in \mathbb{R}$, the following key result relates singular values of $G(r,\theta)$ 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, starting when Byers introduced the first algorithm to compute the distance to instability in 1988 [7].

THEOREM 2.1. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, r, \theta \in \mathbb{R}$ with $r \neq 0$. Then $\gamma \geq 0$ is a singular value of $G(r,\theta)$ defined in (2.1) if and only if $\mathbf{i}r$ is an eigenvalue of the skew-Hamiltonian-Hamiltonian matrix pencil (M, N_θ), where*

$$M \coloneqq \begin{bmatrix} A & 0 \\ 0 & -A^* \end{bmatrix} \quad \text{and} \quad N_\theta \coloneqq \begin{bmatrix} -\mathbf{i}e^{\mathbf{i}\theta}I & \mathbf{i}\gamma\cos\theta I \\ -\mathbf{i}\gamma\cos\theta I & \mathbf{i}e^{-\mathbf{i}\theta}I \end{bmatrix}, \tag{2.2}$$

N_θ is singular if and only if $|\gamma\cos\theta| = 1$, and (M, N_θ) is regular if $|\gamma\cos\theta| \neq 1$.

Proof. It is easy to verify that M and N_θ are, respectively, Hamiltonian and skew-Hamiltonian, and so (M, N_θ) is an sHH matrix pencil. As N_θ is composed of four blocks of different multiples of the $n \times n$ identity matrix, $\det(N_\theta) = 1 - (\gamma\cos\theta)^2$. Thus, $|\gamma\cos\theta| \neq 1$, i.e., N_θ being nonsingular, is clearly a sufficient condition for

(M, N_θ) to be a regular matrix pencil. Now suppose γ is a singular value of $G(r,\theta)$ with left and right singular vectors u and v. Then the following two equations hold:

$$\gamma \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} G(r,\theta) & 0 \\ 0 & G(r,\theta)^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} \quad \text{and} \quad \gamma r \cos\theta \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r\mathrm{e}^{\mathbf{i}\theta} I - A & 0 \\ 0 & r\mathrm{e}^{-\mathbf{i}\theta} I - A^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}.$$

Multiplying the bottom block by -1 and rearranging terms, this is equivalent to

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

Noting that the matrix on the right multiplied by $-\mathbf{i}$ is N_θ completes the proof. ∎

Remark 2.2. By Theorem 2.1, if a point $(\tilde{r}, \tilde{\theta})$ is in the γ-level set of $g(r,\theta)$ for some $\gamma \geq 0$, then $\mathbf{i}\tilde{r} \in \Lambda(M, N_{\tilde{\theta}})$. Note that the converse is not necessarily true. If $\mathbf{i}\tilde{r}$ is an eigenvalue of the matrix pencil $(M, N_{\tilde{\theta}})$, Theorem 2.1 only states that γ is a singular value of $G(\tilde{r}, \tilde{\theta})$. For point $(\tilde{r}, \tilde{\theta})$ to be in the γ-level set, γ would additionally have to be the smallest singular value of $G(\tilde{r}, \tilde{\theta})$. However, if γ is not the minimum singular value of $G(r,\theta)$, then $(\tilde{r}, \tilde{\theta})$ is instead in some $\hat{\gamma}$-level set of $g(r,\theta)$ with $\hat{\gamma} < \gamma$.

Besides being computationally useful for detecting level-set points, Theorem 2.1 provides a way to show that the γ-level set of $g(r,\theta)$ is bounded for $\gamma \in [0, 1)$.

THEOREM 2.3. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma \in [0,1)$. The γ-level set of $g(r,\theta)$ defined in* (2.1) *is bounded. Moreover, if $\alpha(A) < 0$, the γ-level set is compact.*

Proof. For any point $\tilde{r}\mathrm{e}^{\mathbf{i}\theta}$ in the γ-level set of $g(r,\theta)$, γ is a singular value of $G(\tilde{r},\theta)$. Thus by Theorem 2.1, $\mathbf{i}\tilde{r} \in \Lambda(M, N_\theta)$. Furthermore, all eigenvalues of (M, N_θ) must be finite, as $|\gamma| < 1$ implies that N_θ is always nonsingular. Consider the function $m(\theta) \coloneqq \rho(M, N_\theta)$. By continuity of the spectral radius, $m(\theta)$ must have a finite maximal value $m_\star$ on $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Since $|\mathbf{i}\tilde{r}| \leq m_\star$ must hold, the γ-level set of $g(r,\theta)$ is bounded. If $\alpha(A) < 0$, then $g(r,\theta)$ is infinite on all of the imaginary axis, and so its γ-level set cannot contain purely imaginary values. Thus, the γ-level set must additionally be in the open right half-plane and closed, hence compact. ∎

Our new method will require that zero is not an eigenvalue of (M, N_θ). The following theorem gives the precise conditions to meet this requirement, namely, that zero cannot be an eigenvalue of A.

THEOREM 2.4. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, \theta \in \mathbb{R}$. Then the matrix pencil (M, N_θ) defined by* (2.2) *has zero as an eigenvalue if and only if the matrix A also has zero as an eigenvalue. Consequently, $0 \notin \Lambda(A)$ also ensures that (M, N_θ) is regular.*

Proof. The proof is immediate as $\det(M) = \det(A)\det(-A^*)$, so clearly (M, N_θ) is a regular matrix pencil if $0 \notin \Lambda(A)$. ∎

Given $\gamma \geq \mathcal{K}(A)^{-1}$ and some $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, Theorem 2.1 provides a way to compute all the γ-level set points of $g(r,\theta)$ along the ray emanating from the origin specified by angle $\tilde{\theta}$, namely, via computing all the imaginary eigenvalues of $(M, N_{\tilde{\theta}})$. As (M, N_θ) is an sHH pencil, its eigenvalues are symmetric with respect to the imaginary axis, and structure-preserving eigenvalues solvers such as [1] can be used to ensure that computed imaginary eigenvalues are exactly on the imaginary axis in the presence of rounding errors. While this is desirable for numerical robustness, solving generalized eigenvalue problems is many times more expensive than solving a standard eigenvalue problem of the same dimension and, as we explain later, our new methods often do not need this level of robustness. Thus, since N_θ is generically nonsingular, we now

investigate the condition number of N_θ to ascertain the feasibility of instead computing the eigenvalues of $N_\theta^{-1}M$ via the QR algorithm. We first need the following generic result.

LEMMA 2.5. *Let the matrix* $E = \left[\begin{smallmatrix} aI & bI \\ \overline{b}I & \overline{a}I \end{smallmatrix}\right]$ *with* $a, b \in \mathbb{C}$ *such that* $\det(E) \neq 0$. *Then the condition number of* E *is*

$$\kappa(E) = \frac{|a| + |b|}{||a| - |b||}.$$

Proof. Since E is nonsingular, all its singular values are positive and they are equal to the square roots of the eigenvalues of EE^*. As

$$EE^* = \left[\begin{smallmatrix} (a\overline{a}+b\overline{b})I & 2abI \\ 2\overline{a}\overline{b}I & (a\overline{a}+b\overline{b})I \end{smallmatrix}\right] =: \left[\begin{smallmatrix} cI & dI \\ \overline{d}I & cI \end{smallmatrix}\right],$$

$0 = \det(EE^* - \lambda I) = \lambda^2 - 2c\lambda + (c^2 - |d|^2)$, so the eigenvalues of EE^* are

$$\lambda = c \pm |d| = |a|^2 + |b|^2 \pm 2|ab|.$$

Thus, the singular values of E are $||a| \pm |b||$. ∎

THEOREM 2.6. *Let* $A \in \mathbb{C}^{n\times n}$ *and* $\gamma, \theta \in \mathbb{R}$ *with* $|\gamma \cos\theta| \neq 1$. *Then the spectrum of the matrix pencil* (M, N_θ) *defined by* (2.2) *is equal to the spectrum of*

$$M_\theta := N_\theta^{-1}M = \frac{\mathbf{i}}{1 - (\gamma\cos\theta)^2} \begin{bmatrix} \mathrm{e}^{-\mathbf{i}\theta}A & (\gamma\cos\theta)A^* \\ (\gamma\cos\theta)A & \mathrm{e}^{\mathbf{i}\theta}A^* \end{bmatrix}, \tag{2.3}$$

and if $\gamma \in [0, 1)$, *then* $\max_{\theta\in\mathbb{R}} \kappa(N_\theta) = \frac{1+\gamma}{1-\gamma}$.

Proof. The matrix given in (2.3) simply follows by using the obvious explicit form of N_θ^{-1}, which exists if and only if $|\gamma\cos\theta| \neq 1$, and then evaluating $N_\theta^{-1}M$. Applying Lemma 2.5 to N_θ, we have that $\kappa(N_\theta) = \frac{1+|\gamma\cos\theta|}{|1-|\gamma\cos\theta||}$. It is easy to see that if $\gamma \in [0, 1)$, then $\theta = 0$ is a global maximizer of this ratio, thus completing the proof. ∎

For $\gamma = 0.9$, Theorem 2.6 says that the condition number of N_θ is at most only 19, and $\max_{\theta\in\mathbb{R}} \kappa(N_\theta) \to 1$ monotonically as $\gamma \to 0$. While $\kappa(N_\theta)$ does blow up as $\gamma \to 1$, this is mostly inconsequential, since $\gamma = \mathcal{K}(A)^{-1} \in [0.9, 1]$ corresponds to Kreiss constants between 1 and 1.1. In other words, for almost all matrices of interest, encountered values of γ should be much less than 0.9, and so N_θ will be very well conditioned. Hence, there is generally no numerical concern in computing the eigenvalues of (M, N_θ) via $N_\theta^{-1}M$, except that the imaginary axis symmetry will not be maintained exactly via the standard QR algorithm. As we clarify later, computing the spectrum of (M, N_θ) using an sHH structure-preserving eigensolver can always be done as a backup.

2.2. An interpolation-based globality certificate for $g(r, \theta)$. We are now ready to present our first interpolation-based globality certificate, specifically for (1.11b). Given $\gamma \geq 0$, 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 are about to construct a rather well-behaved continuous function $g_\gamma : (-\frac{\pi}{2}, \frac{\pi}{2}) \mapsto [0, \pi^2]$ such that $g_\gamma(\tilde{\theta}) = 0$ holds whenever the ray from the origin determined by angle $\tilde{\theta}$ intersects the γ-level set of $g(r, \theta)$. Hence, if $g_\gamma(\theta)$ 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 $g_\gamma(\tilde{\theta}) = 0$ provide the directions of the rays that intersect the γ-level, and provided these intersection points are not stationary, they can be used to restart optimization to find better (lower) minimizers of (1.11b). By approximating $g_\gamma(\theta)$ via interpolation, we can then ascertain if it has any zeros.

Keeping in mind that the spectrum of (M, N_θ) as defined by (2.2) is always imaginary-axis symmetric, to accomplish our criteria above, consider

$$g_\gamma(\theta) \coloneqq \min\{\operatorname{Arg}(-\mathbf{i}\lambda)^2 : \lambda \in \Lambda(M, N_\theta), \operatorname{Re}\lambda \le 0\}, \tag{2.4a}$$

$$\mathcal{G}(\gamma) \coloneqq \operatorname{int}\{\theta : g_\gamma(\theta) = 0,\ \theta \in (-\tfrac{\pi}{2}, \tfrac{\pi}{2})\}, \tag{2.4b}$$

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

THEOREM 2.7. *Let $A \in \mathbb{C}^{n\times n}$ with $\alpha(A) \le 0$ and $0 \notin \Lambda(A)$. Then for any $\gamma \ge 0$, the function $g_\gamma(\theta)$ defined in* (2.4a) *has the following properties:*

(i) $g_\gamma(\theta) \ge 0$ *for all* $\theta \in \mathcal{D} \coloneqq (-\frac{\pi}{2}, \frac{\pi}{2})$,
(ii) $g_\gamma(\theta) = 0$ *if and only if there exists* $\mathbf{i}r \in \Lambda(M, N_\theta)$ *with* $r \in \mathbb{R}$ *and* $r > 0$,
(iii) $g_\gamma(\theta)$ *is continuous on its entire domain* $\mathcal{D}$,
(iv) $g_\gamma(\theta)$ *is differentiable at a point* θ *if the eigenvalue* $\lambda \in \Lambda(M, N_\theta)$ *attaining the value of* $g_\gamma(\theta)$ *is unique and simple.*

Furthermore, the following properties hold for the set $\mathcal{G}(\gamma)$ *defined in* (2.4b)*:*

(v) *if* $\mathcal{K}(A)^{-1} < \gamma$, *then* $0 < \mu(\mathcal{G}(\gamma))$,
(vi) $\gamma_1 \le \gamma_2$ *if and only if* $\mu(\mathcal{G}(\gamma_1)) \le \mu(\mathcal{G}(\gamma_2))$,
(vii) $\lim_{\gamma\to\infty} \mu(\mathcal{G}(\gamma)) = \pi$,

where $\mu(\cdot)$ *is the Lebesgue measure on* $\mathbb{R}$.

Proof. We begin with $g_\gamma(\theta)$. The first and second properties hold by construction, since $-\mathbf{i}\lambda$ in (2.4a) is always in the (closed) upper half of the complex plane. The third property is a consequence of the continuity of eigenvalues and our assumption that $0 \notin \Lambda(A)$, which via Theorem 2.4 ensures that 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 $g_\gamma(\theta)$.

We now turn to $\mathcal{G}(\gamma)$, which, since it is defined as an interior, is thus open and measurable. Let $(r_\star, \theta_\star)$ be a global minimizer, i.e., $g(r_\star, \theta_\star) = \mathcal{K}(A)^{-1}$, where $r_\star > 0$ and $\theta_\star \in \mathcal{D}$, and let $\mathcal{L}_\gamma \coloneqq \{(r, \theta) : g(r, \theta) < \gamma, r > 0, \theta \in \mathcal{D})\}$ be a strict lower level set of $g(r, \theta)$. As $\mathcal{L}_\gamma$ is open, there exists an open disk neighborhood $\mathcal{N} \subset \mathcal{L}_\gamma$ about $(r_\star, \theta_\star)$, so let $\mathcal{T}$ denote the (positive-length) interval of angles specifying the rays from the origin that intersect $\mathcal{N}$. Since $g(r, \theta) < \gamma$ for all points in $\mathcal{N}$ and $\lim_{r\to 0^+} g(r, \theta) = \infty$ for any $\theta \in \mathcal{D}$ (as $0 \notin \Lambda(A)$), it follows by continuity of $g(r, \theta)$ that for every $\theta \in \mathcal{T}$ there exists at least one $\tilde{r} \in (0, r_\star)$ such that $g(\tilde{r}, \theta) = \gamma$. Thus, by Theorem 2.1 it follows that $g_\gamma(\theta) = 0$ for all $\theta \in \mathcal{T}$, and as $\mu(\mathcal{T}) > 0$ and $\mathcal{T} \subset \mathcal{G}(\gamma)$, the fifth property holds. The sixth property holds by noting that $\gamma_1 \le \gamma_2$ if and only if $\mathcal{L}_{\gamma_1} \subseteq \mathcal{L}_{\gamma_2}$, which in turn is equivalent to $\mathcal{G}(\gamma_1) \subseteq \mathcal{G}(\gamma_2)$. To see this, consider any ray from the origin that intersects $\operatorname{bd}\mathcal{L}_{\gamma_1}$, say, at point $(\hat{r}, \theta)$. Then $g(\hat{r}, \theta) = \gamma_1$, and so as in the argument for the fifth property, there exists $\tilde{r} \in (0, \hat{r})$ such that $g(\tilde{r}, \theta) = \gamma_2$; hence this ray must intersect $\operatorname{bd}\mathcal{L}_{\gamma_2}$ at $(\tilde{r}, \theta)$. Thus, $g_{\gamma_1}(\theta) = 0$ implies $g_{\gamma_2}(\theta) = 0$, and so $\mathcal{G}(\gamma_1) \subseteq \mathcal{G}(\gamma_2)$. Now suppose $\mathcal{G}(\gamma_1) \supset \mathcal{G}(\gamma_2)$ and let $\theta \in \mathcal{G}(\gamma_1) \setminus \mathcal{G}(\gamma_2)$; hence $g_{\gamma_1}(\theta) = 0$ but $g_{\gamma_2}(\theta) > 0$. Then $g(\hat{r}, \theta) = \gamma_1$ holds for some $\hat{r} > 0$, but $g(r, \theta) \ne \gamma_2$ for all $r \in (0, \infty)$, and so $\gamma_2 < \min_{r>0} g(r, \theta) \le \gamma_1$, a contradiction. For the seventh property, we first note that $\lim_{r\to\infty} g(r, \theta) = \sec\theta \ge 1$; hence for any θ such that $\sec\theta \le \gamma$, there again must exist $\tilde{r} > 0$ such that $g(\tilde{r}, \theta) = \gamma$. Thus, it is clear that $\lim_{\gamma\to\infty} \mu(\mathcal{G}(\gamma)) = \pi$ must hold. ∎

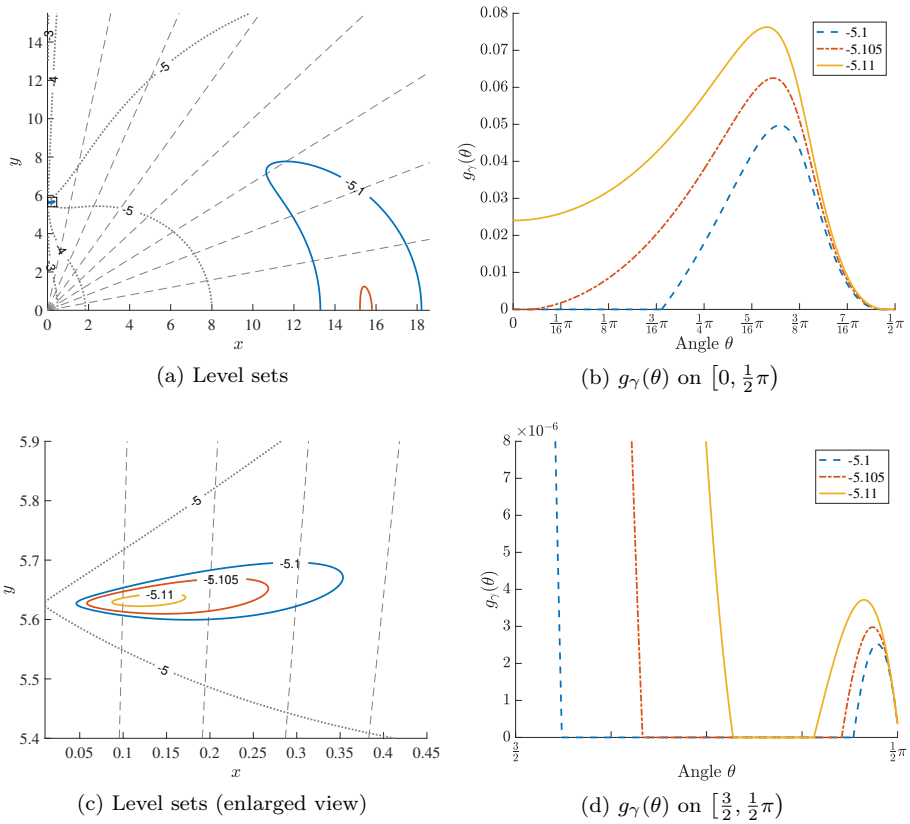
Taken together with Theorem 2.1 and Remark 2.2, it clear that $g_\gamma(\theta)$ meets our new criteria for a level-set test. Given $\gamma \geq 0$, $\tilde{r} > 0$, and some $\tilde{\theta} \in \mathcal{D}$, if point $(\tilde{r}, \tilde{\theta})$ is in the γ-level set of $g(r, \theta)$, then by Theorem 2.1, $\mathbf{i}\tilde{r}$ must be an eigenvalue of matrix pencil $(M, N_{\tilde{\theta}})$ and so $g_\gamma(\tilde{\theta}) = 0$ holds. If $g_\gamma(\tilde{\theta}) = 0$, by definition there exists $\mathbf{i}\tilde{r} \in \Lambda(M, N_{\tilde{\theta}})$ with $r > 0$, and so by Theorem 2.1, γ must be a singular value of $G(\tilde{r}, \tilde{\theta})$. Thus by Remark 2.2, point $(\tilde{r}, \tilde{\theta})$ must either be in the γ-level set of $g(r, \theta)$ or some other $\tilde{\gamma}$-level set with $\tilde{\gamma} < \gamma$. Hence, $g_\gamma(\theta) = 0$ is associated with new starting points for optimization such that a better (lower) minimizer can be found. Finally, if $g_\gamma(\theta) > 0$ for all $\theta \in \mathcal{D}$, then (M, N_θ) has no imaginary eigenvalues on the positive imaginary axis for any $\theta \in \mathcal{D}$, so again by Theorem 2.1, γ is not a singular value of $G(r, \theta)$ for any $r > 0$ and $\theta \in \mathcal{D}$. This in turn means the γ-level set of $g(r, \theta)$ is empty. As $g(r, \theta)$ is continuous, $\gamma < \mathcal{K}(A)^{-1}$ must hold.

Remark 2.8. As we will approximate $g_\gamma(\theta)$ via interpolation, the presence of the square in $\text{Arg}(-\mathbf{i}\lambda)^2$ is to help smooth out the numerically difficult high rate of change that $\text{Arg}(-\mathbf{i}\lambda)$ would otherwise have. To understand this, suppose that the γ-level set of $g(r, \theta)$ consists of a single continuous closed curve enclosing a nonempty convex interior. Then $\mathcal{G}(\gamma) \subset (-\frac{\pi}{2}, \frac{\pi}{2})$ is simply a single interval, and for any θ in $\mathcal{G}(\gamma)$, (M, N_θ) must have two distinct eigenvalues: $\mathbf{i}r_1$ and $\mathbf{i}r_2$ with $r_1, r_2 > 0$. However, as θ approaches either end of interval $\mathcal{G}(\gamma)$, this pair will first coalesce on the imaginary axis and then split apart again, with both eigenvalues moving very rapidly off of the imaginary axis (in opposite directions).

In Figure 1, we show plots of $g_\gamma(\theta)$ for different values of γ for the 10×10 continuous-time example used in [22, section 8]. The example is based on a demo from EigTool [25], 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 $g(r, \theta)$ 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 $g_\gamma(\theta)$ can be reduced to $[0, \frac{\pi}{2})$.

Although we do not know of an analytic way of finding zeros of $g_\gamma(\theta)$, it is a continuous function of one real variable on a fixed finite interval which we can approximate via interpolation. Interpolation-based approximation of $g_\gamma(\theta)$ is practical as $g_\gamma(\theta)$ is rather well behaved on its finite domain and is relatively cheap to evaluate. Moreover, even though $g_\gamma(\theta)$ may be nondifferentiable at some points, modern interpolation software is adept at approximating functions that are nonsmooth and even discontinuous or have singularities. Thus, as finding roots (and extrema) of polynomial or piecewise-polynomial interpolants is easy, approximating $g_\gamma(\theta)$ via interpolation allows a way to find where $g_\gamma(\theta) = 0$ holds, and in turn find γ-level set points of $g(r, \theta)$. Moreover, as we are about to explain, often a high-fidelity approximation for $g_\gamma(\theta)$ is only needed once $\gamma \approx \mathcal{K}(A)^{-1}$ holds, i.e., once our new method has converged. Finally, note that even without interpolation, zeros of $g_\gamma(\theta)$ may be found via sampling, since by Theorem 2.7 $\mu(\mathcal{G}(\gamma)) > 0$ must hold if $\gamma > \mathcal{K}(A)^{-1}$.

In Algorithm 2.1, we provide pseudocode for our new approach to computing continuous-time $\mathcal{K}(A)$ using optimization-with-restarts and our interpolation-based globality certificates, the latter of which we now describe at a high level. Our certificates assume that existing interpolation software can approximate $g_\gamma(\theta)$ to essentially machine precision. Given $\gamma \geq \mathcal{K}(A)^{-1}$, our certificate works by beginning to sample $g_\gamma(\theta)$ for various values of θ in order to approximate it on $\mathcal{D}$, or just $[0, \frac{\pi}{2})$ if A is real. Since interpolation methods are adaptive, this sampling happens in batches, where $g_\gamma(\theta)$ can be evaluated at the requested sample points in an "embarrassingly parallel" manner. If any zeros of $g_\gamma(\theta)$ are encountered during a given batch of



(a) Level sets

(b) $g_\gamma(\theta)$ on $[0, \frac{1}{2}\pi)$

(c) Level sets (enlarged view)

(d) $g_\gamma(\theta)$ on $[\frac{3}{2}, \frac{1}{2}\pi)$

FIG. 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* (1.11b) *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* (1.11b) *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,* $g_\gamma(\theta)$ *for* $\gamma = 10^k$ *is plotted in the right panes, for the respective regions shown in the 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. 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* $g_\gamma(\theta)$ *are zero in the right panes.*

sampling, then γ-level-set points of $g(r, \theta)$ have been detected and the interpolation process is immediately halted. Then the locations of detected level-set points are computed via Theorem 2.1, and these are used to restart optimization in order to find a better (lower) minimizer; for brevity, we assume that the detected level-set points are not exactly stationary. In this case, sampling $g_\gamma(\theta)$ and restarting optimization suffices to lower γ closer to $\mathcal{K}(A)^{-1}$; hence the interpolation process begins anew with the updated version of $g_\gamma(\theta)$. We now consider when interpolation produces a high-fidelity approximation $p_\gamma(\theta)$ for $g_\gamma(\theta)$ but without ever encountering zeros during sampling. To assert that $\gamma = \mathcal{K}(A)^{-1}$ really holds, the interpolant $p_\gamma(\theta)$ is used to check if $g_\gamma(\theta) = 0$ on regions that were not sampled. This is possible to do since, by assumption, $p_\gamma(\theta)$ approximates $g_\gamma(\theta)$ to machine precision. Thus, the global minimizer(s) of the interpolant $p_\gamma(\theta)$ are computed and used to check if $g_\gamma(\theta) = 0$ at

Algorithm 2.1 Interpolation-Based Globality Certificate Algorithm

Input: $A \in \mathbb{C}^{n\times n}$ (nonnormal, $\alpha(A) \le 0$, and $0 \notin \Lambda(A)$) and $z_0 \in \mathbb{C}$ with $\operatorname{Re} z_0 > 0$.
Output: $\gamma^{-1} \approx \mathcal{K}(A)$ (continuous-time).

1: $\mathcal{D} \leftarrow (-\frac{\pi}{2}, \frac{\pi}{2})$
2: **if** A is real **then**
3: $\quad\mathcal{D} \leftarrow [0, \frac{\pi}{2})$
4: **end if**
5: **while** true **do**
6: $\quad\gamma \leftarrow$ computed locally/globally minimal value of (1.11b) initialized from z_0
7: $\quad$*// Begin approximating $g_\gamma(\theta)$ to check convergence or find new starting points*
8: $\quad p_\gamma(\theta) \leftarrow 1$ *// Initial guess for interpolant $p_\gamma(\theta)$ for approximating $g_\gamma(\theta)$*
9: $\quad$**while** $p_\gamma(\theta)$ does not sufficiently approximate $g_\gamma(\theta)$ on $\mathcal{D}$ **do**
10: $\quad\quad[\theta_1, \dots, \theta_l] \leftarrow$ new sample points from $\mathcal{D}$
11: $\quad\quad$*// If new starting points are detected, restart optimization to lower γ:*
12: $\quad\quad$**if** $g_\gamma(\theta_j) = 0$ for some $j \in \{1, \dots, l\}$ **then**
13: $\quad\quad\quad z_0 \leftarrow$ a point $re^{\mathbf{i}\theta_j}$ such that $\mathbf{i}r \in \Lambda(M, N_{\theta_j})$ defined in (2.2) with $r > 0$
14: $\quad\quad\quad$**goto line 6** *// Restart optimization from z_0*
15: $\quad\quad$**end if**
16: $\quad\quad$*// Otherwise, no starting points detected, keep improving $p_\gamma(\theta)$:*
17: $\quad\quad p_\gamma(\theta) \leftarrow$ improved interpolant of $g_\gamma(\theta)$ via $\theta_1, \dots, \theta_l$
18: $\quad$**end while**
19: $\quad$*// $p_\gamma(\theta)$ approximates $g_\gamma(\theta)$ well and no new starting points were encountered*
20: $\quad$*// However, do a final check before asserting that $g_\gamma(\theta)$ has no other zeros:*
21: $\quad[\theta_1, \dots, \theta_l] = \arg\min p_\gamma(\theta)$
22: $\quad$**if** $g_\gamma(\theta_j) = 0$ for some $j \in \{1, \dots, l\}$ **then**
23: $\quad\quad z_0 \leftarrow$ a point $re^{\mathbf{i}\theta_j}$ such that $\mathbf{i}r \in \Lambda(M, N_{\theta_j})$ defined in (2.2) with $r > 0$
24: $\quad\quad$**goto line 6** *// Restart optimization from z_0*
25: $\quad$**else**
26: $\quad\quad$**return** *// $p_\gamma(\theta) \approx g_\gamma(\theta)$ and $\implies$ $\gamma \approx \mathcal{K}(A)^{-1}$*
27: $\quad$**end if**
28: **end while**

NOTE: *For simplicity of the pseudocode, we assume that optimization converges to local/global minimizers exactly and z_0 computed in lines* 13 *and* 23 *is never a stationary point of* (1.11b). *Lines* 7–19 *describe the core of the interpolation-based globality certificate, where we assume the interpolation process for approximating $g_\gamma(\theta)$ is done via some reliable method, e.g., Chebfun. In lines* 20–27, *where a final check is done before asserting convergence, one can additionally/alternatively compute the roots $\{\theta_1, \dots, \theta_l\}$ of $p_\gamma(\theta)$ and check the value of $g_\gamma(\theta)$ at $0.5(\theta_j + \theta_{j+1})$ for all $j = 1, \dots, l-1$.*

these angles. If this yields *newly detected* nonstationary level-set points (since this may simply recover known minimizers that were computed in the last round of optimization), then optimization is restarted to lower γ further. If still no roots of $g_\gamma(\theta)$ are discovered, then the roots of $p_\gamma(\theta)$ are computed and are similarly used to check if $g_\gamma(\theta) = 0$ holds elsewhere, specifically by evaluating $g_\gamma(\theta)$ at the midpoints between consecutive roots. Again, if new level-set points are detected, optimization is restarted from them. Otherwise, our certificate has built a high-fidelity approximation to $g_\gamma(\theta)$ and asserts that it cannot find nonstationary points in the γ-level set. As $\mu(G(\theta)) > 0$ must hold if $\gamma > \mathcal{K}(A)^{-1}$ by Theorem 2.7, the algorithm concludes with $\gamma = \mathcal{K}(A)^{-1}$.

Since the main cost of evaluating $g_\gamma(\theta)$ is computing the spectrum of (M, N_θ), Algorithm 2.1 has a work complexity of $\mathcal{O}(kn^3)$ and a memory complexity of $\mathcal{O}(n^2)$, where k is the total number of function values of $g_\gamma(\theta)$ incurred (over all values of γ encountered). The cost of finding minimizers of (1.11b), the other major component of Algorithm 2.1, can be ignored, as (quasi-)Newton methods only require a handful of iterations to converge (for *two*-variable problems), while evaluating the function value and gradient/Hessian of the objective function in (1.11b) can be done with at most $\mathcal{O}(n^3)$ work and $\mathcal{O}(n^2)$ memory; for more details, see our comments in the introduction on this. As we show in the experiments, when $\gamma > \mathcal{K}(A)^{-1}$, very few samples are needed before a restart occurs. Meanwhile, when $\gamma = \mathcal{K}(A)^{-1}$, the number of interpolation points needed to build a high-fidelity approximation to $g_\gamma(\theta)$ is not necessarily dependent on n, and in fact often acts more like a constant, albeit a large one. The combination of k not being too large and that only a single high-fidelity approximation to $g_\gamma(\theta)$ is typically needed means that our interpolation-based globality certificates can be orders of magnitude faster than earlier techniques based on solving fewer but *much* larger eigenvalue problems. Moreover, using parallel processing for the sampling phases only improves upon this already large performance difference. Finally, in stark contrast to all but one of the methods discussed in subsections 1.1 and 1.2, our new level-set approach does not crucially rely on any *single* computation for correctness. The only way our new interpolation-based approach can fail to restart optimization is if rounding errors prevent detection of level-set points for *every* sampled root of $g_\gamma(\theta)$; as $\mu(G(\theta)) > 0$ holds when $\gamma > \mathcal{K}(A)^{-1}$, this seems quite unlikely and so our new approach is more numerically reliable than previous ones.

Remark 2.9. Note that our interpolation-based globality certificates have 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 (1.9b) may be anywhere in $[0, \infty)$ and may occur on a very fast time scale, which can make finding such maximizers very difficult. Here, $g_\gamma(\theta)$ 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 $g_\gamma(\theta)$ should be substantially easier than finding global maximizers of (1.9b). Second, $g_\gamma(\theta)$ is more reliable to compute and cheaper to obtain; computing $\alpha_\varepsilon(A)$ via the criss-cross algorithms of [5] or [3] often involves computing all eigenvalues of several $2n \times 2n$ matrices.

Remark 2.10. Certainly our certificate function defined in (2.4a) 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 $g_\gamma(\theta)$ 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 $g_\gamma(\theta)$ would be exactly zero if none of the interpolation points happen to fall in $\mathcal{G}(\gamma)$, which may be small when γ is close to $\mathcal{K}(A)^{-1}$. Defining $g_\gamma(\theta)$ so that it generally varies with θ helps to ensure that the function is sufficiently sampled.

2.3. Efficient and robust evaluation of $g_\gamma(\theta)$. By using an sHH structure-preserving eigensolver to compute $\Lambda(M, N_\theta)$, computed imaginary eigenvalues will have exactly zero real part, and so roots of $g_\gamma(\theta)$ should generally be computed as exact roots, i.e., $g_\gamma(\theta)$ should be exactly zero numerically whenever θ corresponds to a ray intersecting the γ-level set. While this is clearly appealing, as mentioned in subsection 2.1, the downside of structure preservation is that it involves solving a generalized

eigenvalue problem, which, if N_θ is nonsingular, is many times slower than solving the equivalent standard eigenvalue problem $N_\theta^{-1}M$. However, in our new algorithm, the vast majority of evaluations for $g_\gamma(\theta)$ will be for values that are nowhere close to being roots. This is because at $\gamma = \mathcal{K}(A)^{-1}$, we expect that $\mu(\mathcal{G}(\theta)) = 0$. Furthermore, if $\gamma > \mathcal{K}(A)^{-1}$, $\mu(\mathcal{G}(\theta)) > 0$ holds, and so some rounding error in the computed eigenvalues can typically be tolerated in obtaining roots of $g_\gamma(\theta)$. Consequently, the increased numerical robustness from using structure-preserving eigensolvers is actually often not relevant in our new certificates. With this in mind, we propose the following way to evaluate $g_\gamma(\theta)$ much faster while still maintaining numerical reliability.

Given $\gamma \geq \mathcal{K}(A)^{-1}$ and $\tilde{\theta} \in \mathcal{D}$, if $N_{\tilde{\theta}}$ is singular, then $g_\gamma(\tilde{\theta})$ must be evaluated by computing the spectrum of the matrix pencil $(M, N_{\tilde{\theta}})$, so in this case there is little reason not to use a structure-preserving eigensolver. However, if $N_{\tilde{\theta}}$ is nonsingular, then $g_\gamma(\tilde{\theta})$ is initially evaluated via computing the eigenvalues of $N_{\tilde{\theta}}^{-1}M$, which, as we have established in subsection 2.1, is not an issue as $\kappa(N_\theta)$ is typically small. Given some small tolerance $\texttt{tol} > 0$, if $g_\gamma(\tilde{\theta})$ is not attained by an eigenvalue λ such that $\min\{|\mathrm{Im}\,\lambda|, |\lambda|\} \leq \texttt{tol}$, i.e., λ is deemed not too close to the positive imaginary axis, then $g_\gamma(\tilde{\theta})$ can be considered to have been computed with sufficient accuracy to assert that angle $\tilde{\theta}$ is indeed not a root. Otherwise, the eigenvalues near the positive imaginary axis are, via Theorem 2.1, used to check if level-set points to restart optimization have been detected. If so, then this is sufficient. The only case that remains is that eigenvalues near the positive imaginary axis have been computed but level-set points have not been detected. As not detecting level-set points *could* be the result of rounding errors, $\Lambda(M, N_{\tilde{\theta}})$ is now recomputed using a structure-preserving eigensolver to either overcome any rounding errors or verify that indeed $g_\gamma(\tilde{\theta}) > 0$ holds.

As we expect $\mu(G(\theta)) = 0$ to hold once $\gamma = \mathcal{K}(A)^{-1}$, only a small minority of evaluations of $g_\gamma(\theta)$ should require the additional computation with the structure-preserving eigensolver. As such, the overall running time of our new algorithm should be much faster than if the structure-preserving eigensolver were always used, and by construction, numerical reliability remains uncompromised.

3. A new approach for computing discrete-time Kreiss constants. We now adapt our new globality certificates to compute discrete-time Kreiss constants to arbitrary accuracy, i.e., to find global minimizers of (1.11a). To do this, we will adapt Algorithm 2.1 and develop a new interpolation-based globality certificate for discrete-time $\mathcal{K}(A)$. In this discrete-time setting, a polar parametrization is used for both finding (feasible) minimizers of (1.11a) (see [22, section 3.2] for details) and the interpolation-based globality certificate itself. Thus, consider

$$h(r,\theta) \coloneqq \sigma_{\min}(H(r,\theta)) \quad \text{and} \quad H(r,\theta) \coloneqq \frac{r\mathrm{e}^{\mathrm{i}\theta}I - A}{r-1}, \tag{3.1}$$

so

$$\mathcal{K}(A)^{-1} = \inf_{r>1,\, \theta\in(-\pi,\pi]} h(r,\theta).$$

To create a discrete-time $\mathcal{K}(A)$ analogue of $g_\gamma(\theta)$, we make the following assumptions. If A is normal or $\rho(A) > 1$, computing $\mathcal{K}(A)$ is trivial, so we assume that neither condition holds. Also, while in the previous section $g_\gamma(\theta)$ required that $0 \notin \Lambda(A)$, our new certificate for discrete-time $\mathcal{K}(A)$ requires that $\gamma^2 \notin \Lambda(AA^*)$.

3.1. Level sets of $h(r,\theta)$ and another 1D radial level-set test. Since a key part of interpolation-based globality certificates is a 1D radial level-set test, we

begin with an analogue of Theorem 2.1.

THEOREM 3.1. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, r, \theta \in \mathbb{R}$ with $r \neq 1$. Then $\gamma \geq 0$ is a singular value of $H(r,\theta)$ defined in* (3.1) *if and only if $\mathbf{i}r$ is an eigenvalue of the skew-Hamiltonian-Hamiltonian matrix pencil (S, T_θ), where*

$$S \coloneqq \begin{bmatrix} A & -\gamma I \\ \gamma I & -A^* \end{bmatrix} \quad \text{and} \quad T_\theta \coloneqq \begin{bmatrix} -\mathbf{i}\mathrm{e}^{\mathbf{i}\theta} I & \mathbf{i}\gamma I \\ -\mathbf{i}\gamma I & \mathbf{i}\mathrm{e}^{-\mathbf{i}\theta} I \end{bmatrix}, \tag{3.2}$$

T_θ is singular if and only if $|\gamma| = 1$, and (S, T_θ) is regular if $|\gamma| \neq 1$.

Proof. It is easy to verify that S and T_θ are, respectively, Hamiltonian and skew-Hamiltonian, and so (S, T_θ) is an sHH matrix pencil. Furthermore, the determinant of T_θ is simply $\det(T_\theta) = 1 - \gamma^2$. Thus, $|\gamma| \neq 1$, i.e., T_θ being nonsingular, is sufficient for (S, T_θ) to be a regular matrix pencil. Now suppose γ is a singular value of $H(r,\theta)$ with left and right singular vectors u and v. Then the following two equations hold:

$$\gamma \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} H(r,\theta) & 0 \\ 0 & H(r,\theta)^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} \text{ and } \gamma(r-1) \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r\mathrm{e}^{\mathbf{i}\theta} I - A & 0 \\ 0 & r\mathrm{e}^{-\mathbf{i}\theta} I - A^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}.$$

Rearranging terms, this is equivalent to

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

Combining terms and multiplying the bottom block row by -1, we equivalently have

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

Noting that the matrix on the right multiplied by $-\mathbf{i}$ is T_θ completes the proof. ∎

The point of Remark 2.2, with appropriate substitutions, similarly applies to Theorem 3.1, $h(r,\theta)$, and $H(r,\theta)$, and Theorem 3.1 also allows a way to show that the γ-level set of $h(r,\theta)$ is bounded for $\gamma \in [0, 1)$.

THEOREM 3.2. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma \in [0, 1)$. The γ-level set of $h(r,\theta)$ defined in* (3.1) *is bounded. Moreover, if $\rho(A) < 1$, the γ-level set is compact.*

Proof. The proof follows similarly to the proof of Theorem 2.3, with the key part being that the eigenvalues of (S, T_θ) are finite for all θ (since T_θ is nonsingular if $\gamma \neq \pm 1$), and so $\max_{\theta\in\mathbb{R}} \rho(S, T_\theta)$ must be finite. ∎

As with our continuous-time certificate using $g_\gamma(\theta)$ and (M, N_θ), for discrete-time $\mathcal{K}(A)$ we need to ensure that zero cannot be an eigenvalue of (S, T_θ), the precise conditions for which are given by the following result.

THEOREM 3.3. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, \theta \in \mathbb{R}$. Then the matrix pencil (S, T_θ) defined by* (3.2) *has zero as an eigenvalue if and only if the matrix AA^* has γ^2 as an eigenvalue. Consequently, $\gamma^2 \notin \Lambda(AA^*)$ also ensures that (S, T_θ) is regular.*

Proof. As the blocks of S are all $n \times n$ and the lower two blocks γI and $-A^*$ commute, the if-and-only-if equivalence holds because $\det(S) = \det(-AA^* + \gamma^2 I)$. Lastly, if $0 \notin \Lambda(S)$, then clearly (S, T_θ) must be a regular matrix pencil. ∎

Finally, we consider the condition number of T_θ and when computing the eigenvalues of (S, T_θ) via $T_\theta^{-1} S$ is possible. The following result shows that $\kappa(T_\theta) = \max_{\theta\in\mathbb{R}} \kappa(N_\theta)$, i.e., T_θ will generally be very well conditioned for all relevant values of γ, and hence using $T_\theta^{-1} S$ to compute $\Lambda(S, T_\theta)$ is not problematic.

THEOREM 3.4. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, \theta \in \mathbb{R}$ with $\gamma \neq \pm 1$. Then the spectrum of the matrix pencil (S, T_θ) defined by* (3.2) *is equal to the spectrum of*

$$S_\theta \coloneqq T_\theta^{-1} S = \frac{\mathbf{i}}{1-\gamma^2} \begin{bmatrix} \mathrm{e}^{-\mathbf{i}\theta} A - \gamma^2 I & \gamma(A^* - \mathrm{e}^{-\mathbf{i}\theta} I) \\ \gamma(A - \mathrm{e}^{\mathbf{i}\theta} I) & \mathrm{e}^{\mathbf{i}\theta} A^* - \gamma^2 I \end{bmatrix}, \tag{3.3}$$

and if $\gamma \in [0, 1)$, then $\max_{\theta\in\mathbb{R}} \kappa(T_\theta) = \frac{1+\gamma}{1-\gamma}$.

Proof. The matrix given in (3.3) simply follows by using the explicit form of S_θ^{-1}, which exists if and only if $\gamma \neq \pm 1$. Applying Lemma 2.5 to T_θ with $\gamma \in [0, 1)$, $\kappa(T_\theta) = \frac{1+\gamma}{1-\gamma}$, thus completing the proof. ∎

3.2. An interpolation-based globality certificate for $h(r, \theta)$. For (1.11a), we correspondingly consider sweeping the entire complex via a ray from the origin to see where it intersects the γ-level set of $h(r, \theta)$ outside of the closed unit disk. Thus, we construct a new continuous function $h_\gamma : (-\pi, \pi] \mapsto [0, \pi^2]$ similar to (2.4a),

$$h_\gamma(\theta) \coloneqq \min\{\mathrm{Arg}(-\mathbf{i}\lambda)^2 : \lambda \in \Lambda(S, T_\theta), \lambda \notin [0, \mathbf{i}], \mathrm{Re}\,\lambda \leq 0\}, \tag{3.4a}$$

$$\mathcal{H}(\gamma) \coloneqq \mathrm{int}\{\theta : h_\gamma(\theta) = 0,\ \theta \in (-\pi, \pi]\}, \tag{3.4b}$$

similarly keeping in mind that $\Lambda(S, T_\theta)$ always has imaginary-axis symmetry, regardless of whether or not the level sets of $h(r, \theta)$ have symmetry.

THEOREM 3.5. *Let $A \in \mathbb{C}^{n\times n}$ with $\rho(A) \leq 1$. Then for any $\gamma \geq 0$ such that $\gamma^2 \notin \Lambda(AA^*)$, the function $h_\gamma(\theta)$ defined in* (3.4a) *has the following properties:*

(i) *$h_\gamma(\theta) \geq 0$ for all $\theta \in \mathcal{D} \coloneqq (-\pi, \pi]$,*
(ii) *$h_\gamma(\theta) = 0$ if and only if there exists $\mathbf{i}r \in \Lambda(S, T_\theta)$ with $r \in \mathbb{R}$ and $r > 1$,*
(iii) *$h_\gamma(\theta)$ is continuous on its entire domain $\mathcal{D}$,*
(iv) *$h_\gamma(\theta)$ is differentiable at a point θ if the eigenvalue $\lambda \in \Lambda(S, T_\theta)$ attaining the value of $h_\gamma(\theta)$ is unique and simple.*

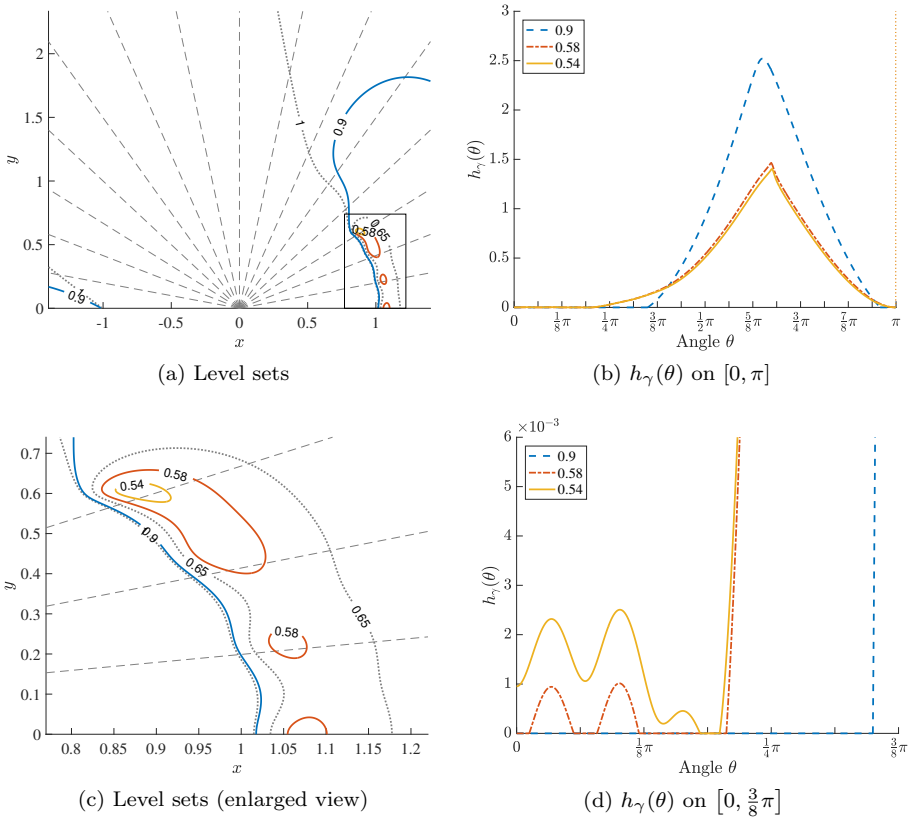
Furthermore, the following properties hold for the set $\mathcal{H}(\gamma)$ defined in (3.4b)*:*

(v) *if $\mathcal{K}(A)^{-1} < \gamma$, then $0 < \mu(\mathcal{H}(\gamma))$,*
(vi) *$\gamma_1 \leq \gamma_2$ if and only if $\mu(\mathcal{H}(\gamma_1)) \leq \mu(\mathcal{H}(\gamma_2))$,*
(vii) *if $\gamma > 1$, then $\mu(\mathcal{H}(\gamma)) = 2\pi$,*

where $\mu(\cdot)$ is the Lebesgue measure on $\mathbb{R}$.

Proof. The proof mostly follows the proof of Theorem 2.7, now using Theorems 3.1 and 3.3 instead of Theorems 2.1 and 2.4. The notable differences are as follows. The second property requires the exclusion of any imaginary eigenvalues of (S, T_θ) that are also in the interval $[0, \mathbf{i}]$, per the definition of $h_\gamma(\theta)$ given in (3.4a). This key change keeps $h_\gamma(\theta)$ strictly positive whenever (S, T_θ) has one or more eigenvalues on the imaginary axis in $[0, \mathbf{i}]$ but not in $(\mathbf{i}, \infty)$. The continuity property is unaffected by this exclusion but does require our assumption that $\gamma^2 \notin \Lambda(AA^*)$, which by Theorem 3.3 guarantees that zero is never an eigenvalue of (S, T_θ) for any $\theta \in \mathbb{R}$. For the properties of $\mathcal{H}(\gamma)$, the main differences to note are that $\lim_{r\to\infty} h(r, \theta) = 1$ for any θ, while $\lim_{r\to 1^+} h(r, \theta) = \infty$ for almost all θ. This second limit can only be finite for at most n values of $\theta \in \mathcal{D}$, namely, at angles corresponding to unimodular eigenvalues of A. ∎

In Figure 2, we show plots of $h_\gamma(\theta)$ for different values of γ for the 10×10 discrete-time example used in [22, section 8], namely, $A = \frac{1}{13}B + \frac{11}{10}I$, where matrix $B =$ `convdiff_demo(11)` from EigTool. As A is real, the level sets of $h(r, \theta)$ are symmetric with respect to the real axis, and so only the upper half of the complex plane is shown.



(a) Level sets (b) $h_\gamma(\theta)$ on $[0, \pi]$

(c) Level sets (enlarged view) (d) $h_\gamma(\theta)$ on $[0, \frac{3}{8}\pi]$

FIG. 2. *The top left pane shows a contour plot of the level sets (now in linear scale) of the objective function in* (1.11a) *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* (1.11a) *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* $h_\gamma(\theta)$ *function is plotted in the right panes, for the respective regions shown in the 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. The correspondence between the* γ*-level set and where* $h_\gamma(\theta) = 0$ *is clearly evident. In the top right pane, the discontinuity in* $h_\gamma(\theta)$ *for* $\gamma = 0.54$ *near* $\theta = \pi$ *is due to excluding eigenvalues of* (S, T_θ) *that lie inside the eccentric ellipse* $\delta^{-2}x^2 + y^2 = 1$ *with* $\delta = 10^{-8}$.

The removal of eigenvalues from $\Lambda(S, T_\theta)$ on the imaginary axis in the interval $[0, \mathbf{i}]$ for $h_\gamma(\theta)$ requires further comment. In fact, any eigenvalue of (S, T_θ) inside the closed unit disk is irrelevant, but per Remark 2.2, Theorem 3.1 may detect level-set points of $\sigma_k(zI - A)/|(|z| - 1)|$, for any $k = 1, \ldots, n$, where $\sigma_k(\cdot)$ is the kth largest singular value. However, excluding all eigenvalues in the unit disk could introduce discontinuities, as $\mathrm{Arg}(\cdot)$ may jump if an eigenvalue of (S, T_θ) enters or exits the unit disk. Instead, by only excluding those in $[0, \mathbf{i}]$, continuity is preserved, and while $h_\gamma(\theta)$ may become infinitesimal as eigenvalues inside the unit disk may be arbitrarily close to $[0, \mathbf{i}]$, they cannot introduce a zero of $h_\gamma(\theta)$ precisely since $[0, \mathbf{i}]$ itself is excluded. In practice, using a structure-preserving eigensolver for (S, T_θ) means that computed eigenvalues on the imaginary axis will be exactly imaginary, and so removing any that are in $[0, \mathbf{i}]$ can be done exactly. When a structure-preserving eigensolver is

not used, two other measures can help deal with rounding errors in the real parts of computed eigenvalues. First, in lines 13 and 23 for the adaptation of Algorithm 2.1 (see subsection 3.3 for the complete details), we need only consider eigenvalues $\mathbf{i}r$ of (S, T_θ) with $r > 1$ so that optimization is restarted only if *feasible* nonstationary points are found; for $h_\gamma(\theta)$, this crucial specification ensures that $h_\gamma(\theta)$ is increasingly better approximated by interpolant $p_\gamma(\theta)$ until either $p_\gamma(\theta) \approx h_\gamma(\theta)$ or a detected zero of $h_\gamma(\theta)$ *also* leads to detection of a level-set point outside the unit disk. Second, to help ensure that $h_\gamma(\theta) = 0$ only if $re^{\mathbf{i}\theta}$ with $r > 1$ is a level-set point, we can discard any eigenvalue of (S, T_θ) that lies inside the eccentric ellipse defined by $\delta^{-2}x^2 + y^2 = 1$ for some small $\delta > 0$. Technically, this may still introduce discontinuities in $h_\gamma(\theta)$, but they are much less likely to occur than when excluding the entire unit disk ($\delta = 1$). Note that such a discontinuity can be seen in Figure 2; see the caption for details.

3.3. Adapting Algorithm 2.1 for discrete-time $\mathcal{K}(A)$. The following modifications of Algorithm 2.1 are needed for computing discrete-time $\mathcal{K}(A)$. For input, it is assumed that A is nonnormal with $\rho(A) \leq 1$ and $z_0 \in \mathbb{C}$ with $|z_0| > 1$. While $h_\gamma(\theta)$ requires that $\gamma^2 \notin \Lambda(AA^*)$, per Theorem 3.3, this is a very mild assumption; clearly there are only up to n values of $\gamma \geq 0$ such that $\gamma^2 \in \Lambda(AA^*)$, and in the unlikely case that one of these is encountered, simply perturbing γ by a slight amount would suffice. In lines 1–3, $\mathcal{D}$ should be initially set to $(-\pi, \pi]$ and reduced to $[0, \pi]$ if A is real. Throughout the pseudocode and accompanying note, (1.11b) and $g_\gamma(\theta)$ should be replaced by (1.11a) and $h_\gamma(\theta)$, respectively, and $0.5(\theta_l + (\theta_1 + 2\pi))$ should also be included when doing the additional check described in the note. As alluded to earlier, in lines 13 and 23, "$\mathbf{i}r \in \Lambda(M, N_{\theta_j})$ defined in (2.2) with $r > 0$" should be replaced with "$\mathbf{i}r \in \Lambda(S, T_{\theta_j})$ defined in (3.2) with $r > 1$." For increased efficiency, $h_\gamma(\theta)$ should be evaluated in an analogous manner as described in subsection 2.3 for $g_\gamma(\theta)$, but in this case it is only necessary to consider recomputing the eigenvalues of $T_\theta^{-1}S$ via its matrix pencil form when a computed eigenvalue is within a distance `tol` of the interval $[\mathbf{i}, \infty)$ on the imaginary axis (as opposed to the interval $[0, \infty)$). These modifications for discrete-time $\mathcal{K}(A)$ do not alter the $\mathcal{O}(kn^3)$ work complexity and $\mathcal{O}(n^2)$ memory characteristics.

4. A new approach for computing the distance to uncontrollability. We now turn to adapting our new globality certificates for $\tau(A, B)$, i.e., to find global minimizers of (1.2). For the optimization phases, local minimizers of (1.2) should be found using Cartesian coordinates, with a quasi-Newton method or Newton's method for fast local convergence; the gradient of (1.2) can be found in [6, p. 358], while the corresponding Hessian can be obtained via a straightforward modification to the derivation of the Hessian of (1.11b) in [22, section 3.1]. For our interpolation-based globality certificate for $\tau(A, B)$, we again use polar coordinates:

$$f(r, \theta) \coloneqq \sigma_{\min}(F(r, \theta)) \quad \text{and} \quad F(r, \theta) \coloneqq \begin{bmatrix} A - re^{\mathbf{i}\theta} I & B \end{bmatrix}, \tag{4.1}$$

so

$$\tau(A, B) = \inf_{r \geq 0,\, \theta \in (-\pi, \pi]} f(r, \theta).$$

Our upcoming $\tau(A, B)$ analogue of $g_\gamma(\theta)$ and $h_\gamma(\theta)$ requires that $\gamma^2 \notin \Lambda(AA^* + BB^*)$, a condition which can be easily ensured, as we will explain.

4.1. Level sets of $f(r, \theta)$ and another 1D radial level-set test. We again begin with a result enabling a radial 1D level-set test. Our following result is essentially a modified version of the result derived in [14, eqn. (2.4)]. As noted there,

similar results were also previously developed in [8, Theorem 3.1] and [13, Lemmas 2.1 and 2.2]. The key difference here, besides some simplifications, is that we derive an sHH matrix pencil for the desirable imaginary-axis symmetry of its eigenvalues. The proof is also a bit different.

THEOREM 4.1. *Let $A \in \mathbb{C}^{n\times n}$ and $\gamma, r, \theta \in \mathbb{R}$ with $\gamma \neq 0$. Then $\gamma > 0$ is a singular value of $F(r,\theta)$ defined in* (4.1) *if and only if $\mathbf{i}r$ is an eigenvalue of the regular skew-Hamiltonian-Hamiltonian matrix pencil (C, D_θ), where*

$$C := \begin{bmatrix} A & \widetilde{B} \\ \gamma I & -A^* \end{bmatrix} \quad \text{and} \quad D_\theta := \begin{bmatrix} -\mathbf{i}\mathrm{e}^{\mathbf{i}\theta} I & 0 \\ 0 & \mathbf{i}\mathrm{e}^{-\mathbf{i}\theta} I \end{bmatrix}, \tag{4.2}$$

$\widetilde{B} := \frac{1}{\gamma} BB^* - \gamma I$, and D_θ is always nonsingular.

Proof. Clearly D_θ is always nonsingular and it easy to verify that it is also skew-Hamiltonian and C is Hamiltonian; hence, (C, D_θ) is a regular sHH matrix pencil. Now suppose γ is a singular value of $F(r,\theta)$ with left and right singular vectors u and $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$. Then the following two equations hold:

$$\gamma \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} F(r,\theta) & 0 \\ 0 & F(r,\theta)^* \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} \text{ and } \gamma \begin{bmatrix} u \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} A - r\mathrm{e}^{\mathbf{i}\theta} I & B & 0 \\ 0 & 0 & A^* - r\mathrm{e}^{-\mathbf{i}\theta} I \\ 0 & 0 & B^* \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ u \end{bmatrix}.$$

The last block row yields $v_2 = \frac{1}{\gamma} B^* u$, and so by making this substitution in the equation above, we equivalently have

$$\gamma \begin{bmatrix} u \\ v_1 \end{bmatrix} = \begin{bmatrix} A - r\mathrm{e}^{\mathbf{i}\theta} I & \frac{1}{\gamma} BB^* & 0 \\ 0 & 0 & A^* - r\mathrm{e}^{-\mathbf{i}\theta} I \end{bmatrix} \begin{bmatrix} v_1 \\ u \\ u \end{bmatrix} = \begin{bmatrix} A - r\mathrm{e}^{\mathbf{i}\theta} I & \frac{1}{\gamma} BB^* \\ 0 & A^* - r\mathrm{e}^{-\mathbf{i}\theta} I \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix}.$$

Rearranging terms to separate out the terms involving r and multiplying the resulting lower block row by -1 yields

$$\begin{bmatrix} A & \frac{1}{\gamma} BB^* \\ 0 & -A^* \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix} - \gamma \begin{bmatrix} u \\ -v_1 \end{bmatrix} = \begin{bmatrix} A & \widetilde{B} \\ \gamma I & -A^* \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix} = r \begin{bmatrix} \mathrm{e}^{\mathbf{i}\theta} I & 0 \\ 0 & -\mathrm{e}^{-\mathbf{i}\theta} I \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix}.$$

Noting that the matrix on the right multiplied by $-\mathbf{i}$ is D_θ completes the proof. ∎

Remark 2.2, with appropriate substitutions, also applies to Theorem 4.1, $f(r,\theta)$, and $F(r,\theta)$. While Theorem 4.1 can also be used to show that the γ-level set of $f(r,\theta)$ is compact for any finite γ, this is not necessary as it is already well known; see [6, p. 353]. A third argument for this is via [26, Theorem 2.2] and equivalently considering $\sigma_{\min}(\left[\begin{smallmatrix} A - zI & B \end{smallmatrix}\right]^*)$, whose lower level sets are rectangular pseudospectra.

THEOREM 4.2. *Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and $\gamma \geq 0$. The γ-level set of $f(r,\theta)$ defined in* (4.1) *is compact.*

While in the Kreiss constant setting it is clear when the level sets of $g(r,\theta)$ and $h(r,\theta)$ are symmetric with respect to the real axis, the symmetry conditions for $f(r,\theta)$ are slightly more nuanced.

THEOREM 4.3. *Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and let $\sigma_k(\cdot)$ denote the kth singular value. 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 $\sigma_k(M) \iff \sigma_k^2 \in \Lambda(MM^*)$:

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

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

We now consider the conditions under which zero is an eigenvalue of (C, D_θ), since our interpolation-based globality certificates require excluding this possibility.

THEOREM 4.4. *Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and $\gamma, \theta \in \mathbb{R}$ with $\gamma \neq 0$. Then the matrix pencil (C, D_θ) defined by (4.2) has zero as an eigenvalue if and only if the matrix $AA^* + BB^*$ has γ^2 as an eigenvalue.*

Proof. Since the blocks of C are all square matrices of the same size and the lower two blocks γI and $-A^*$ commute, the if-and-only-if equivalence holds because

$$\begin{aligned}
\det(C) = \det(-AA^* - \gamma\widetilde{B}) &= \det(-AA^* - \gamma(\tfrac{1}{\gamma}BB^* - \gamma I)) \\
&= \det(-AA^* - BB^* + \gamma^2 I).
\end{aligned}$$
∎

As the next result states, it is clear that $D_\theta^{-1}C$ can be used to compute the eigenvalues of (C, D_θ) when forgoing structure-preserving eigensolvers.

THEOREM 4.5. *Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and $\gamma, \theta \in \mathbb{R}$. The condition number of D_θ, $\kappa(D_\theta)$, equals one for any θ, and the spectrum of matrix pencil (C, D_θ) defined by (4.2) is equal to the spectrum of*

$$C_\theta \coloneqq D_\theta^{-1}C = \mathbf{i}\begin{bmatrix} \mathrm{e}^{-\mathbf{i}\theta}A & \mathrm{e}^{-\mathbf{i}\theta}\widetilde{B} \\ -\gamma\mathrm{e}^{\mathbf{i}\theta}I & \mathrm{e}^{\mathbf{i}\theta}A^*\end{bmatrix}. \tag{4.3}$$

Proof. The proof is an immediate consequence of the fact that D_θ is a unitary diagonal matrix for any θ. ∎

4.2. An interpolation-based globality certificate for $f(r, \theta)$. Unlike for Kreiss constants, there are no domain restrictions for where a minimizer of (1.2) may lie, so our $\tau(A, B)$ certificate must sweep the entire complex plane with a ray from the origin to find level-set points. Moreover, the origin can be immediately tested by simply evaluating $f(0, \theta)$ for any θ. Thus, given $\gamma > 0$, we construct another continuous function $f_\gamma : (-\pi, \pi] \mapsto [0, \pi^2]$ similar to $g_\gamma(\theta)$:

$$f_\gamma(\theta) \coloneqq \min\{\mathrm{Arg}(-\mathbf{i}\lambda)^2 : \lambda \in \Lambda(C, D_\theta), \mathrm{Re}\,\lambda \leq 0\}, \tag{4.4a}$$

$$\mathcal{F}(\gamma) \coloneqq \mathrm{int}\{\theta : f_\gamma(\theta) = 0,\ \theta \in (-\pi, \pi]\}, \tag{4.4b}$$

noting that $\Lambda(C, D_\theta)$ always has imaginary-axis symmetry.

THEOREM 4.6. *Let $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$. Then for any $\gamma > 0$ such that $\gamma^2 \notin \Lambda(AA^* + BB^*)$, the function $f_\gamma(\theta)$ defined in* (4.4a) *has the following properties:*

(i) *$f_\gamma(\theta) \geq 0$ for all $\theta \in \mathcal{D} := (-\pi, \pi]$,*
(ii) *$f_\gamma(\theta) = 0$ if and only if there exists $\mathbf{i}r \in \Lambda(C, D_\theta)$ with $r \in \mathbb{R}$ and $r > 0$,*
(iii) *$f_\gamma(\theta)$ is continuous on its entire domain $\mathcal{D}$,*
(iv) *$f_\gamma(\theta)$ is differentiable at a point θ if the eigenvalue $\lambda \in \Lambda(C, D_\theta)$ attaining the value of $f_\gamma(\theta)$ is unique and simple.*

Furthermore, the following properties hold for the set $\mathcal{F}(\gamma)$ defined in (4.4b)*:*

(v) *if $\tau(A, B) < \gamma$, then $0 < \mu(\mathcal{F}(\gamma))$,*
(vi) *$\gamma_1 \leq \gamma_2$ if and only if $\mu(\mathcal{F}(\gamma_1)) \leq \mu(\mathcal{F}(\gamma_2))$,*
(vii) *if $\gamma > f(0, \theta)$ for any $\theta \in \mathbb{R}$, then $\mu(\mathcal{F}(\gamma)) = 2\pi$,*

where $\mu(\cdot)$ is the Lebesgue measure on $\mathbb{R}$.

Proof. This proof also follows the proof of Theorem 2.7, now using Theorems 4.1 and 4.4 instead of Theorems 2.1 and 2.4. Here the continuity property of $f_\gamma(\theta)$ requires our assumption that $\gamma^2 \notin \Lambda(AA^* + BB^*)$, which by Theorem 4.4 guarantees that zero is never an eigenvalue of (C, D_θ) for any $\theta \in \mathbb{R}$. For $\mathcal{F}(\gamma)$, the corresponding arguments use that $f(r, \theta)$ is continuous and $\lim_{r\to\infty} f(r, \theta) = \infty$ for any θ. ∎

Note that by Theorems 4.1 and 4.4, our assumption that γ^2 is not an eigenvalue of $AA^* + BB^*$ is equivalent to γ not being a singular value of $F(0, \theta)$ for any $\theta \in \mathbb{R}$. As such, the properties of $f_\gamma(\theta)$ hold as long as $\gamma < f(0, \theta)$. Since optimization-with-restarts monotonically decreases the value of γ until it converges to $\tau(A, B)$, we can easily guarantee that γ^2 is never an eigenvalue of $AA^* + BB^*$ just by initializing at the origin. Provided the origin is not a stationary point, optimization guarantees finding a point $(\tilde{r}, \tilde{\theta})$ such that $f(\tilde{r}, \tilde{\theta}) < f(0, \theta)$. Otherwise, either other starting points can be evaluated in order to find a function value lower than $f(0, \theta)$ or the initial value of γ can simply be set to slightly less than $f(0, \theta)$ before commencing the first certification computation. Note that while $f_\gamma(\theta)$ is not defined for $\gamma = 0$, this is not a problem as there is no need to do a globality check when $f(r, \theta) = 0$, as $f(r, \theta)$ is never negative. Finally, if $f(r, \theta) = f(r, -\theta)$, i.e., the level sets have real-axis symmetry, then the domain $\mathcal{D}$ can be reduced to $[0, \pi]$.

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

4.3. Adapting Algorithm 2.1 for $\tau(A, B)$. We modify Algorithm 2.1 to compute $\tau(A, B)$ as follows. For input, $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, and $z_0 \in \mathbb{C}$ without restriction. By also including the origin as an initial point, ensuring $\gamma^2 \notin \Lambda(AA^* + BB^*)$ only requires that the origin not be stationary or, if no other starting points result in a value of γ less than $f(0, \theta)$, that the initial value of γ be set slightly less than $f(0, \theta)$. In lines 1–3, $\mathcal{D}$ should be initially set to $(-\pi, \pi]$ and reduced to $[0, \pi]$ if the level sets have real-axis symmetry, per the conditions given in Theorem 4.3. Throughout the pseudocode and accompanying note, (1.11b) and $g_\gamma(\theta)$ should be replaced by (1.2) and $f_\gamma(\theta)$, respectively, and $0.5(\theta_l + (\theta_1 + 2\pi))$ should also be included when doing the additional check described in the note. In lines 13 and 23, "$\mathbf{i}r \in \Lambda(M, N_{\theta_j})$ defined in (2.2) with $r > 0$" should be replaced with "$\mathbf{i}r \in \Lambda(C, D_{\theta_j})$ defined in (4.2) with $r > 0$." For increased efficiency, $f_\gamma(\theta)$ should be evaluated in a manner analogous to that described in subsection 2.3 for $g_\gamma(\theta)$. The $\mathcal{O}(kn^3)$ work complexity and $\mathcal{O}(n^2)$ memory characteristics again hold.

5. Numerical experiments. To validate our new interpolation-based globality certificates, we implemented a proof-of-concept of Algorithm 2.1 (and its variants

described in sections 3 and 4; for simplicity, in section 5 we refer to any of these as Algorithm 2.1) in MATLAB and compared it against the existing state-of-the-art methods of [22] for continuous- and discrete-time Kreiss constants and of [15] for the distance to uncontrollability. Experiments were performed in MATLAB R2017b using a computer with two Intel Xeon Gold 6130 processors (16 cores each, 32 total) and 192GB of RAM. The supplementary material includes code (kreiss_dtu_code.zip [local/web 7.72MB]), test examples, and a detailed descriptions of both our implementation and experimental setup for reproducibility of all results, tables, and figures in the paper (kreiss_dtu_mitchell_supp.pdf [local/web 170KB]); for brevity, we only give essential details here. We plan to add "production-ready" implementations of Algorithm 2.1 for $\mathcal{K}(A)$ and $\tau(A, B)$ to a future release of ROSTAPACK [20].

For implementing Algorithm 2.1, `fminunc` was used for finding local minimizers, while $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ were evaluated using `eig`. Per subsection 2.3 on efficient evaluation, our code first attempts to compute the values of $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ via the standard eigenvalue problem formulations in (2.3), (3.3), and (4.3). For simplicity, our prototype code only resorts to using the generalized eigenvalue problems in (2.2), (3.2), and (4.2) when infs, nans, or errors are encountered; for a more robust implementation, eigenvalues of these sHH matrix pencils should be computed using a structure-preserving eigensolver such as [1, 4], and this should also be done whenever the computed values of $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ (when using `eig` and the standard eigenvalue problems) are close to zero. For approximating $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$, we used Chebfun [9], a sophisticated and efficient toolbox for "computing with functions to about 15-digit accuracy"[3] that is also adept at handling nonsmooth functions when its `splitting` option is enabled. To replicate the design of Algorithm 2.1, where optimization is restarted when zeros of $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ are encountered, our code simply throws and catches an error in order to halt Chebfun; this allows us to immediately restart optimization without letting Chebfun finish building a high-fidelity approximation and requires no modifications to Chebfun itself. Our prototype attempts to restart optimization using one or more of the detected level-set points but not necessarily all; a more robust implementation might first check whether or not any of these points are nonstationary before deciding to halt Chebfun early. When Chebfun does build an approximation without encountering zeros, our prototype does the additional global convergence checks described in Algorithm 2.1 (see lines 20–27 and its accompanying note). Finally, our code terminates if none of the new starting points leads to a meaningful decrease in the estimate γ, i.e., if the relative improvement in γ is less than 10^{-14}. This additional test is necessary in practice as optimization software will generally not compute minimizers exactly and our interpolation-based globality certificates may still detect level-set points when a global minimizer has been found to numerical precision.

Remark 5.1. To demonstrate our interpolation-based globality certificates and encourage multiple restarts, we intentionally chose starting points such that only local, not global, minimizers would be found on the first round of optimization. Moreover, our prototype performs a new globality certificate as soon as optimization results in a relative decrease of 10^{-6} or more from the value of γ for the preceding certificate. As such, some detected level-set points may be ignored, which, if used, could have led to larger decreases in γ. In practice, it will likely be more efficient to run optimization from all or at least many of the detected level-set points, to avoid making unnecessarily

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

small updates. For similar reasons, more than a single starting point should be used.

Remark 5.2. For conceptual simplicity, we have so far intentionally omitted a few other notable implementation details, which we now briefly describe. First, in Algorithm 2.1, to account for rounding error, the interpolation-based globality certificates should not be done with the value of γ computed in line 6 but rather $(1-\texttt{tol})\gamma$ for a relative tolerance $\texttt{tol} \in (0,1)$, e.g., $\texttt{tol} = 10^{-14}$. Second, for continuous-time $\mathcal{K}(A)$ and $\tau(A,B)$, when the level sets do not have real-axis symmetry, it may be beneficial to shift the "search point" from the origin; e.g., for $\mathcal{K}(A)$, one might instead use the average of the imaginary parts of the eigenvalues of A. Finally, for discrete-time $\mathcal{K}(A)$ and $\tau(A,B)$, there is an additional technique that can often provide an additional factor-of-two speedup. By noting that both $h_\gamma(\theta)$ and $h_\gamma(\theta+\pi)$ can be computed via a single computation of the spectrum of (S, T_θ), our interpolation-based globality certificates can be computed by approximating $\min\{h_\gamma(\theta), h_\gamma(\theta+\pi)\}$ over half of the domain $\mathcal{D}$, and the same can be analogously done when computing $\tau(A,B)$. For simplicity in the comparisons here, we forgo using this additional optimization.

5.1. Comparisons to earlier methods. Since we address parallel computation in subsection 5.2, 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. The test problems, whose dimensions are listed in Table 1, are as follows. For continuous-time $\mathcal{K}(A)$, `companion` (stab.) is the stabilized EigTool example we used to generate Figure 1, while `boeing('S')` and `orrsommerfeld` are directly from EigTool. For discrete-time $\mathcal{K}(A)$, `convdiff` (mod.) is the modified EigTool example we used to generate Figure 2, while `randn` #1 (stab.) and `randn` #2 (stab.) are randomly generated stable complex matrices, scaled so their spectral radii are 0.999. While these discrete-time examples have very low Kreiss constants, they are useful for demonstration as $h(r,\theta)$ has multiple different local minima for each of them. For $\tau(A,B)$, [15, section 4.3] used real-valued examples generated by setting A to different sizes of the `kahan` demo from EigTool and $B = \texttt{randn}(n,m)$; for our experiments here, we generated two such examples using larger values of n and m, namely, `kahan` ($m = 20$) and `kahan` ($m = 30$). Of the eight examples, `randn` #1 (stab.) and `randn` #2 (stab.) do not have level sets with real-axis symmetry, while the others do.

For computing Kreiss constants, we compare the efficiency of Algorithm 2.1 with the earlier 2D level-set methods of [22], using the code provided in the supplementary material of [22]. However, the running times we report in Table 1 are *not* for the complete algorithms of [22], but rather just the time to perform a single 2D level-set test. Recall that the methods of [22] always require performing at least one 2D level-set test, and these tests are the dominant cost, with $\mathcal{O}(n^6)$ work when using dense eigensolvers. Thus, it suffices to time a single level-set test for each method of [22]. We did not use the asymptotically faster divide-and-conquer versions from [22], as they appear to be less reliable when computing Kreiss constants; see [22, section 8]. For continuous-time $\mathcal{K}(A)$, the generalized eigenvalue problems that appear in the fixed- and variable-distance 2D level-set tests of [22] were solved with `eig`, while the corresponding quadratic eigenvalue problems for the discrete-time $\mathcal{K}(A)$ tests were solved with `polyeig`. In Table 1, we see that for the small ($n = 10$) examples, `companion` (stab.) and `convdiff` (mod.), the total running time of Algorithm 2.1 is comparable with the cost of a single 2D level-set of [22], but our new approach is much faster for larger dimensions. In fact, for the other continuous-time $\mathcal{K}(A)$ examples, `boeing('S')` ($n = 55$) and `orrsommerfeld` ($n = 100$), Algorithm 2.1 is generally over 1000 times faster than a single 2D level-set test. For the two `randn`-based discrete-time

TABLE 1

The eight problems tested. The size of the matrix A is given by n, while z_0 is the initial point used for the first round of optimization. The values of $\mathcal{K}(A)$ and $\tau(A,B)$ computed by Algorithm 2.1 are given under "Computed Value." Elapsed wall-clock times (in seconds) are given in the three rightmost columns. For Algorithm 2.1, the total running times are reported under "New." For Kreiss constants, rather than running the complete algorithms of [22], we only recorded the time to perform a single 2D level-set test ("Single LS Test" in the table) for each problem. Consequently, these times greatly underreport the actual costs to run the full algorithms of [22]. As the methods of [22] use either fixed- or variable-distance 2D level-set tests, times are reported for both types, respectively under "2D Fixed" and "2D Vari.," except for `randn` *#2 (stab.), where out-of-memory errors occurred. For $\tau(A,B)$, for which only fixed-distance 2D level-set tests are relevant, the times to compute $\tau(A,B)$ using the complete divide-and-conquer method of [15] ("Full D&C Alg." in the table) are reported.*

Problem	n	z_0	Computed Value	Time (sec.) New	2D Fixed	2D Vari.
$\mathcal{K}(A)$ (continuous)					Single LS Test	
`companion` (stab.)	10	6+6**i**	$1.29186707013556 \times 10^{5}$	0.5	0.5	1.0
`boeing('S')`	55	1+50**i**	$3.62541052800213 \times 10^{4}$	6.1	6226.5	3446.7
`orrsommerfeld`	100	10+10**i**	$3.93230474282055 \times 10^{1}$	149.6	170547.2	197426.0
$\mathcal{K}(A)$ (discrete)					Single LS Test	
`convdiff` (mod.)	10	−1+1**i**	$1.89501339090580 \times 10^{0}$	1.8	0.9	0.9
`randn` #1 (stab.)	50	1+1**i**	$1.75843606578311 \times 10^{0}$	128.7	3324.0	3248.0
`randn` #2 (stab.)	100	1−1**i**	$2.35849495574647 \times 10^{0}$	1223.8	— `out-of-mem` —	
$\tau(A,B)$					Full D&C Alg.	
`kahan` ($m=20$)	60	0+0**i**	$3.88211512261161 \times 10^{-2}$	51.1	246.2	—
`kahan` ($m=30$)	150	0+0**i**	$1.82581469530120 \times 10^{-2}$	644.4	27454.4	—

$\mathcal{K}(A)$ examples, Algorithm 2.1 is roughly 25 times faster than a single 2D level-set test for $n = 50$, while it was not even possible to time the discrete-time 2D level-set tests for $n = 100$, since `polyeig` immediately ran out of memory (on a computer with a 192GB of RAM). To compare accuracy of the methods, we only considered the two small examples (both $n = 10$), due to the high cost of running the 2D level-set-based methods for larger n. The $\mathcal{K}(A)$ estimates computed by Algorithm 2.1 in Table 1 for `companion` (stab.) and `convdiff` (mod.) agree, respectively, to 11 and 15 digits to the corresponding values reported in [22, Table 1] for the optimization-with-restart methods of [22]. The slight discrepancy for `companion` (stab.) is almost certainly due to the fact that optimization solvers do not find minimizers exactly, and so there will generally be some variability in the least significant digits of γ. This can likely be dealt with via tighter tolerances, using different optimization solvers, and/or using more starting points per restart.

For computing the distance to uncontrollability, we compared Algorithm 2.1 with the divide-and-conquer-based method of [15]. Since divide-and-conquer has an asymptotic work complexity of $\mathcal{O}(n^4)$ on average and $\mathcal{O}(n^5)$ in the worst case, it was feasible to run the full method of [15] on our test problems; specifically, we compared Algorithm 2.1 against the `dist_uncont_hybrid` routine,[4] which uses BFGS for optimization and divide-and-conquer 2D level-set tests when `opts.method=1` and `opts.eig_method=1` are set. For the smaller `kahan`-based example ($n = 60$, $m = 20$), Algorithm 2.1 is 4.8 times faster than `dist_uncont_hybrid`, while for the larger ex-

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

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 $g_\gamma(\theta)$, $h_\gamma(\theta)$, or $f_\gamma(\theta)$ for the current estimate γ 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. Note that for `convdiff` *(mod.), the last certificate actually produced new starting points, but optimization was unable to meaningfully lower estimate γ further, and so our code terminated after a round of optimization instead of after a certificate test.*

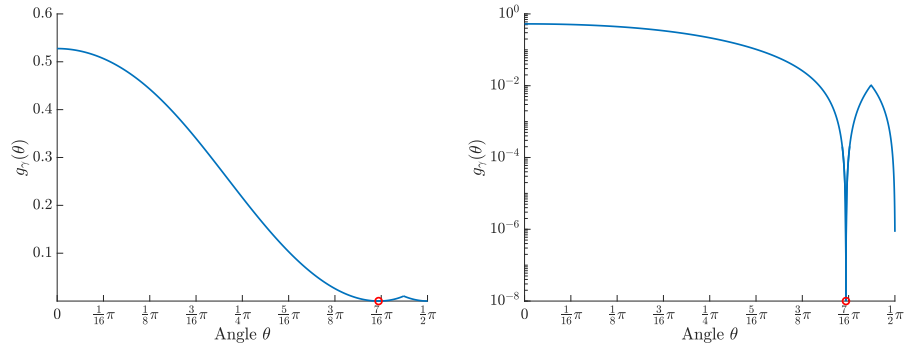
	# of θ's evaluated per certificate and rel. diff. in γ							
Problem	Restart 1		Restart 2		Restart 3		Restart 4	
`companion` (stab.)	15	1e-02	**389**	—	—	—	—	—
`boeing('S')`	15	7e-01	15	7e-01	**535**	—	—	—
`orrsommerfeld`	15	9e-01	**3048**	—	—	—	—	—
`convdiff` (mod.)	15	3e-01	15	4e-02	31	3e-02	**4084**	1e-15
`randn` #1 (stab.)	63	1e-01	**12448**	—	—	—	—	—
`randn` #2 (stab.)	15	2e-01	15	2e-01	127	1e-01	**18672**	—
`kahan` ($m = 20$)	15	7e-01	**3529**	—	—	—	—	—
`kahan` ($m = 30$)	15	6e-01	15	2e-01	**6246**	—	—	—

ample ($n = 150$, $m = 30$), Algorithm 2.1 is 42.6 times faster. We expect that this performance gap will generally widen more as n increases. The $\tau(A, B)$ estimates computed by Algorithm 2.1 for `kahan` ($m = 20$) and `kahan` ($m = 30$) agreed, respectively, to 12 and 13 digits with those computed by `dist_uncont_hybrid`, with our new method returning the (slightly) smaller answers for both.

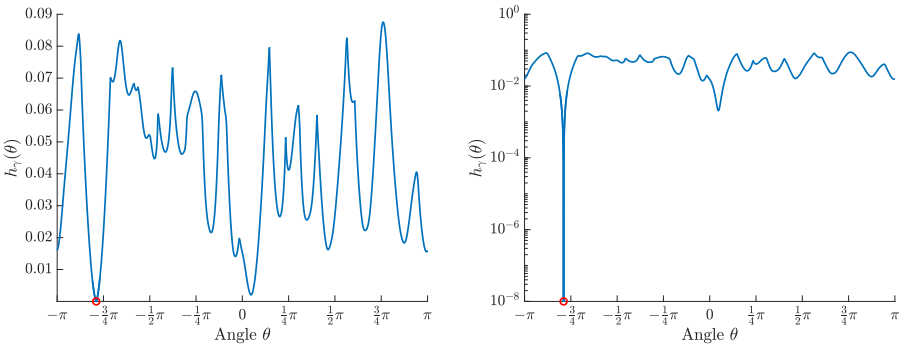
In Table 2, we show the number of points at which Chebfun evaluates $g_\gamma(\theta)$, $h_\gamma(\theta)$, or $f_\gamma(\theta)$ (as appropriate) for each interpolation-based globality certificate that is performed. As can be seen, before a global minimizer is obtained, relatively few values of θ are evaluated by Chebfun before new starting points are discovered and optimization commences again, demonstrating that high-fidelity approximations are indeed only needed once a global minimizer has been found. Furthermore, as hoped, the number of function evaluations needed to build the final interpolants does not dramatically increase as the problems get larger. The number of function evaluations is instead correlated with how complex $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ are, which is not necessarily related to the problem dimension. In Figure 3, we plot $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ for the final values of γ computed by Algorithm 2.1 for three of our examples.

5.2. Additional acceleration via parallel processing. The main components of Algorithm 2.1 are "embarrassingly parallel." Optimization can be run from multiple starting points in parallel, to hopefully find a global minimizer on any given iteration without increasing runtime. For our interpolation-based globality certificates, any time Chebfun provides a vector of different values of θ, obtaining the corresponding function values of $g_\gamma(\theta)$, $h_\gamma(\theta)$, or $f_\gamma(\theta)$ is also "embarrassingly parallel." We focus on this latter task, as it is the dominant cost and is dependent on the average size of vectors provided by Chebfun. To obtain speedup data, we recomputed the final certificates for our three largest problems, where `parpool(cores)` was called with `cores` set to 2, 4, 8, 16, and 32, and the values of $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ were computed inside a `parfor` loop. We did these tests with the Chebfun preference `'min_samples'` retained at its default value of 17 and with it increased to 65, comparing speedups with respect to our single-core configuration used in subsection 5.1.

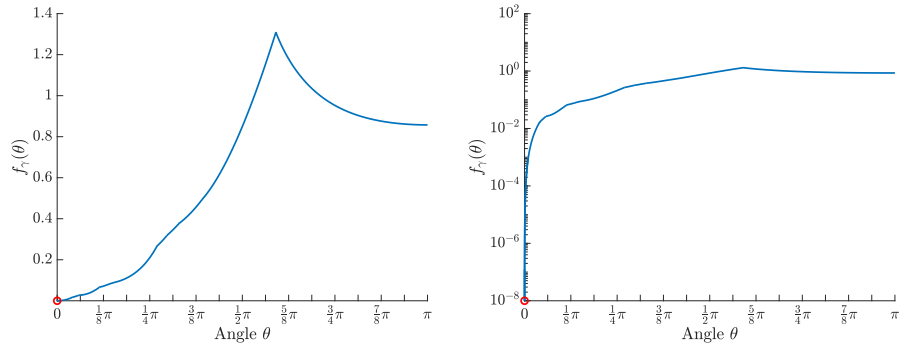
In Table 3, the best speedups range from 6.6 to 9.1, a significant boost. While



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



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



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

FIG. 3. *The top two plots show $g_\gamma(\theta)$ at the final value of γ computed by our new method for the* `boeing('S')` *example, in linear and $\log_{10}$ scales. The circle denotes the angle of the best minimizer obtained by optimization and corresponds to the single place where $g_\gamma(\theta) = 0$ (which is more easily seen in the $\log_{10}$ plot on the right), confirming that γ is the globally minimal value. The same is done for $h_\gamma(\theta)$ and* `randn` *#2 in the middle plots and for $f_\gamma(\theta)$ and* `kahan` *($m = 30$) in the bottom plots.*

this is not high utilization of 32 cores, the average number of θ values provided at a time by Chebfun, which was often about 15 to 20, is an upper limit on achievable speedup. As reported in the # column of Table 3, the parallel region of our code is entered and exited hundreds of times, which comes with a very high overhead.

Since varying `'min_samples'` had little impact on performance and the total num-

TABLE 3

Speedups with respect to the number of θ's evaluated per second while Chebfun is building the final interpolant for the three largest problems; the reason speedups are not with respect to overall running time is because the total number of function evaluations Chebfun needed was not always the same as the number of cores was changed. The last two columns, "#" and "Avg. Size," show, respectively, 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 exiting the `parfor` *loop many times in order for Chebfun to evaluate more and more points.*

	Speedup per # of cores					Vector of θ's	
Problem	2	4	8	16	32	#	Avg. Size
Chebfun `min_samples`: 17							
`orrsommerfeld`	2.4	3.6	4.9	6.4	5.7	184	16.6
`randn` #2 (stab.)	2.9	4.5	6.3	8.5	9.0	797	21.8
`kahan` ($m = 30$)	2.6	3.9	5.5	7.7	8.6	430	14.5
Chebfun `min_samples`: 65							
`orrsommerfeld`	2.6	3.8	5.3	6.6	6.1	162	20.1
`randn` #2 (stab.)	3.0	4.9	6.9	9.1	8.8	608	29.3
`kahan` ($m = 30$)	2.7	4.3	6.2	8.3	8.8	385	17.3

ber of vectors, we analyzed the Chebfun code to determine how its amenability to parallelization might be improved. 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 and does 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, as 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 by a concatenation of them. For each piece, a final safety test for accuracy (`@chebtech/sampleTest.m`) is done by evaluating a pair of *hard-coded* points in the interval the piece is approximating over. This too can keep the average vector length low and increase the total number of vectors. For parallel processing, it would be more efficient to speculatively evaluate these two fixed values for each piece by batching them in with the first vector of initial sample points and storing this pair of function values for recall later.

Remark 5.3. Parallel eigensolvers such as [2] could also be used to accelerate solving the large eigenvalue problems in the 2D level-set tests of [22] and [14], but this would not reduce their high memory requirements, nor does it seem likely that this would be competitive with our interpolation-based certificates even using serial computation, let alone parallel computation.

6. Concluding remarks. We have seen that our new interpolation-based globality certificates are generally orders of magnitudes more efficient than the existing techniques of [22] for Kreiss constants and those of [14, 15] for the distance to uncontrollability. While our new approach assumes that $g_\gamma(\theta)$, $h_\gamma(\theta)$, and $f_\gamma(\theta)$ are adequately sampled to find their zeros, this seems a rather mild assumption in practice, as per Theorems 2.7, 3.5, and 4.6, they will be zero on sets of *positive measure* before a global minimizer has been obtained. The nature of our adequate interpolation assumption is quite different from the exact arithmetic assumption used in the earlier methods of [14, 15, 22], and we believe it to be a more pragmatic choice, both

in terms of efficiency and reliability. Finally, while in this paper we have considered the three specific problems of computing continuous-time Kreiss constants, discrete-time Kreiss constants, and the distance to uncontrollability, we again emphasize that our new approach of interpolation-based globality certificates is for general global optimization problems of singular value functions in two real variables. In fact, after submitting this manuscript, we have since used the idea of interpolation-based globality certificates to obtain a new algorithm for computing "sep-lambda" [21] that is much faster than the method of [16]. However, there are many fundamental differences in this case, in both the nature of the associated global optimization problem and our resulting algorithm.

Acknowledgments. The author is very grateful to Michael L. Overton for supporting several research visits to the Courant Institute in New York and for many helpful comments on this manuscript. The author also thanks the anonymous referees for reviewing the paper and providing helpful feedback.

REFERENCES

[1] P. Benner, R. Byers, V. Mehrmann, and H. Xu, *Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 165–190, https://doi.org/10.1137/S0895479800367439.

[2] P. Benner, M. Köhler, and J. Saak, *Fast approximate solution of the non-symmetric generalized eigenvalue problem on multicore architectures*, in Parallel Computing: Accelerating Computational Science and Engineering (CSE), M. Bader, A. Bodeand, H.-J. Bungartz, M. Gerndt, G. R. Joubert, and F. Peters, eds., Adv. Parallel Comput. 25, IOS Press, 2014, pp. 143–152, https://doi.org/10.3233/978-1-61499-381-0-143.

[3] P. Benner and T. Mitchell, *Extended and improved criss-cross algorithms for computing the spectral value set abscissa and radius*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1325–1352, https://doi.org/10.1137/19M1246213.

[4] P. Benner, V. Sima, and M. Voigt, *Algorithm* 961: *Fortran* 77 *subroutines for the solution of skew-Hamiltonian/Hamiltonian eigenproblems*, ACM Trans. Math. Software, 42 (2016), 24, https://doi.org/10.1145/2818313.

[5] J. V. Burke, A. S. Lewis, and M. L. Overton, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 359–375, https://doi.org/10.1093/imanum/23.3.359.

[6] J. V. Burke, A. S. Lewis, and M. L. Overton, *Pseudospectral components and the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 350–361, https://doi.org/10.1137/S0895479803433313.

[7] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881, https://doi.org/10.1137/0909059.

[8] R. Byers, *Detecting nearly uncontrollable pairs*, in Signal Processing, Scattering and Operator Theory, and Numerical Methods, Vol. III, M. A. Kaashoek, J. H. Schuppen, and A. C. M. Ran, eds., Birkhäuser, Boston, MA, 1990, pp. 447–457.

[9] T. A. Driscoll, N. Hale, and L. N. Trefethen, *Chebfun Guide*, Pafnuty, Oxford, UK, 2014, http://www.chebfun.org/docs/guide/.

[10] R. Eising, *Between controllable and uncontrollable*, Systems Control Lett., 4 (1984), pp. 263–264, https://doi.org/10.1016/S0167-6911(84)80035-3.

[11] M. Embree and B. Keeler, *Pseudospectra of matrix pencils for transient analysis of differential-algebraic equations*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1028–1054, https://doi.org/10.1137/15M1055012.

[12] A. Fazzi, N. Guglielmi, and I. Markovsky, *Computing common factors of matrix polynomials with applications in system and control theory*, in Proceedings of the 58th IEEE Conference on Decision and Control (CDC), Nice, France, 2019, pp. 7721–7726, https://doi.org/10.1109/CDC40024.2019.9030137.

[13] M. Gao and M. Neumann, *A global minimum search algorithm for estimating the distance to uncontrollability*, Linear Algebra Appl., 188/189 (1993), pp. 305–350, https://doi.org/10.1016/0024-3795(93)90472-Z.

[14] M. Gu, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 989–1003, https://doi.org/10.1137/S0895479897328856.
[15] 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 (2006), pp. 477–502, https://doi.org/10.1137/05063060X.
[16] M. Gu and M. L. Overton, *An algorithm to compute* Sep_λ, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 348–359, https://doi.org/10.1137/050622584.
[17] H.-O. Kreiss, *Über die Stabilitätsdefinition für Differenzengleichungen die partielle Differentialgleichungen approximieren*, BIT, 2 (1962), pp. 153–181, https://doi.org/10.1007/bf01957330.
[18] I. Markovsky, A. Fazzi, and N. Guglielmi, *Applications of polynomial common factor computation in signal processing*, in Latent Variable Analysis and Signal Separation, Y. Deville, S. Gannot, R. Mason, M. D. Plumbley, and D. Ward, eds., Lecture Notes in Comput. Sci. 10891, Springer, Cham, 2018, pp. 99–106, https://doi.org/10.1007/978-3-319-93764-9_10.
[19] E. Mengi, *Measures for Robust Stability and Controllability*, Ph.D. thesis, New York University, New York, 2006, https://cs.nyu.edu/media/publications/mengi_emre.pdf.
[20] T. Mitchell, *ROSTAPACK: RObust STAbility PACKage*, http://timmitchell.com/software/ROSTAPACK.
[21] T. Mitchell, *Fast Computation of* sep_λ *via Interpolation-Based Globality Certificates*, preprint, https://arxiv.org/abs/1911.05136, 2019.
[22] T. Mitchell, *Computing the Kreiss constant of a matrix*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 1944–1975, https://doi.org/10.1137/19M1275127.
[23] C. Paige, *Properties of numerical algorithms related to computing controllability*, IEEE Trans. Automat. Control, 26 (1981), pp. 130–138.
[24] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.
[25] T. G. Wright, *EigTool*, http://www.comlab.ox.ac.uk/pseudospectra/eigtool/, 2002.
[26] T. G. Wright and L. N. Trefethen, *Pseudospectra of rectangular matrices*, IMA J. Numer. Anal., 22 (2002), pp. 501–519, https://doi.org/10.1093/imanum/22.4.501.