Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Certifying Global Optimality for the $\mathcal{L}_\infty$-Norm Computation of Large-Scale Descriptor Systems ⋆

P. Schwerdtner * **E. Mengi** ** **M. Voigt** *,***

* *Technische Universität Berlin, Institut für Mathematik, FG Numerische Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany*
** *Koç University, Department of Mathematics, Rumeli Feneri Yolu 34450, Sarıyer, Istanbul, Turkey*
*** *University of Hamburg, Department of Mathematics, Center for Optimization and Approximation, Bundesstraße 55, 20146 Hamburg, Germany*

Abstract: We present a method for the certification of algorithms that approximate the $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norm of transfer functions of large-scale (descriptor) systems. This certification is needed because such algorithms depend heavily on user input, and may converge to a local maximizer of the related singular value function leading to an incorrect value, much lower than the actual norm. Hence, we design an algorithm that determines whether a given value is less than the $\mathcal{L}_\infty$-norm of the transfer function under consideration, and that does not require user input other than the system matrices. In the algorithm, we check whether a certain structured matrix pencil has any purely imaginary eigenvalues by repeatedly applying a structure-preserving shift-and-invert Arnoldi iteration combined with an appropriate shifting strategy.
Our algorithm consists of two stages. First, an interval on the imaginary axis which may contain imaginary eigenvalues is determined. Then, in the second stage, a shift is chosen on this interval and the eigenvalues closest to this shift are computed. If none of these eigenvalues is purely imaginary, then an imaginary interval around the shift of appropriate length is removed such that two subintervals remain. This second stage is then repeated on the remaining two subintervals until either a purely imaginary eigenvalue is found or no critical subintervals are left.
We show the effectiveness of our method by testing it without any parameter adaptation on a benchmark collection of large-scale systems.

Keywords: Computer-aided control systems design, descriptor systems, eigenvalue problems, $\mathcal{H}_\infty$ control, linear control systems, norms, numerical algorithms, robust control

1. INTRODUCTION

Consider a real-rational transfer function

$$H(s) := C(sE - A)^{-1}B + D, \tag{1}$$

where $E, A \in \mathbb{R}^{n\times n}$ are such that the pencil $sE - A \in \mathbb{R}[s]^{n\times n}$ is regular, i. e., $\det(sE - A)$ is not the zero polynomial, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$. If H has no poles on the imaginary axis, then the $\mathcal{L}_\infty$-norm of H is defined by

$$\|H\|_{\mathcal{L}_\infty} := \sup_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2 = \sup_{\omega\in\mathbb{R}} \sigma_1(H(\mathrm{i}\omega)),$$

where $\sigma_i(\cdot)$ denotes the ith largest singular value of its matrix argument. If H has no poles with nonnegative real part, then the $\mathcal{L}_\infty$-norm of H coincides with its $\mathcal{H}_\infty$-norm which is defined by

$$\|H\|_{\mathcal{H}_\infty} := \sup_{\lambda\in\mathbb{C}^+} \|H(\lambda)\|_2,$$

where $\mathbb{C}^+ := \{\lambda \in \mathbb{C} \mid \mathrm{Re}(\lambda) > 0\}$.

Well established and globally convergent algorithms for computing the $\mathcal{L}_\infty$-norm from Boyd and Balakrishnan [1990], Bruinsma and Steinbuch [1990] as well as their extension to the case $E \neq I_n$ (where I_n denotes the identity matrix in $\mathbb{R}^{n\times n}$) in Benner et al. [2012] are based on the following fact: Let the pencil $sE - A$ have no imaginary eigenvalues, and define the matrix pencil

$$s\underbrace{\begin{bmatrix} 0 & -E^\mathsf{T} & 0 & 0 \\ E & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{=:N} - \underbrace{\begin{bmatrix} 0 & -A^\mathsf{T} & -C^\mathsf{T} & 0 \\ -A & 0 & 0 & -B \\ -C & 0 & \gamma I_p & -D \\ 0 & -B^\mathsf{T} & -D^\mathsf{T} & \gamma I_m \end{bmatrix}}_{=:M_\gamma} \in \mathbb{R}[s]^{q\times q}, \tag{2}$$

where $q = 2n + m + p$. Then, for $\omega \in \mathbb{R}$, the matrix $\mathrm{i}\omega N - M_\gamma \in \mathbb{C}^{q\times q}$ is singular if and only if there is an $i \in \{1, \ldots, \min(m, p)\}$ such that $\sigma_i(H(\mathrm{i}\omega)) = \gamma$. If we additionally have that $\gamma > \inf_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2$, then $\|H\|_{\mathcal{L}_\infty} < \gamma$ if and only if (2) has no imaginary eigenvalues.

⋆ This work was supported by the German Research Foundation (DFG) within the priority program 1897: "Calm, Smooth, Smart – Novel Approaches for Influencing Vibrations by Means of Deliberately Introduced Dissipation". Furthermore, the first author acknowledges support by the Berlin International Graduate school in Model and Simulation based Research (BIMoS).

In the numerical methods, the latter condition must be checked repeatedly for various values of γ and, at each such step, the computation of all imaginary eigenvalues of $sN - M_\gamma$ is required. Since these pencils are even, that is $N = -N^\mathsf{T}$ and $M_\gamma = M_\gamma^\mathsf{T}$, structured factorization techniques discussed in Benner et al. [2016] can be used to compute these imaginary eigenvalues in a reliable manner. However, since sparsity cannot be exploited with such factorization methods, these are limited to the small and dense case with q being a few thousand at most.

Methods for computing the $\mathcal{L}_\infty$-norm of transfer functions with large sparse matrices E and A have been under consideration only recently. Some are based on optimization over spectral value sets and are only applicable to the $\mathcal{H}_\infty$-norm computation (Guglielmi et al. [2013], Voigt [2015]), others rely on Newton's method (Freitag et al. [2014]), or use model order reduction techniques (Aliyev et al. [2017], Schwerdtner and Voigt [2018]). The latter class of algorithms is even applicable to irrational transfer functions. However, all of these methods converge only to a local maximizer of $\sigma_1(H(\mathrm{i}\cdot))$. In other words, it cannot be guaranteed that what they return is the globally maximal value of $\sigma_1(H(\mathrm{i}\cdot))$.

1.1 Problems with Local Optimality and Need for Global Optimality Certification

We illustrate the problems that arise from convergence to only a local maximizer with `linorm_subsp` [1], an implementation of the algorithm proposed in Aliyev et al. [2017]. This algorithm uses interpolatory model order reduction to reduce the large-scale $\mathcal{L}_\infty$-norm computation problem to a small-scale one, which then can be dealt with using well established and globally convergent methods. It is based on a greedy interpolation strategy; in each iteration, a point on the imaginary axis where the $\mathcal{L}_\infty$-norm of the current reduced transfer function is attained is added to the set of interpolation points, in particular the projection subspaces are expanded accordingly. In case of convergence, this method converges superlinearly to a local maximum of $\sigma_1(H(\mathrm{i}\cdot))$. In many cases, users know the critical interval in which the $\mathcal{L}_\infty$-norm might be attained, and can thus place initial interpolation points there to avoid stagnation at only a local maximum. However, this is not the case in general. This is illustrated in the following example.

The `peec` [2] system has a maximum singular value plot consisting of many sharp peaks such that the greedy interpolation strategy can miss certain peaks entirely. This is shown in Fig. 1. It can be seen that the maximum singular value of the reduced transfer function that is obtained after running `linorm_subsp` does not interpolate the maximum singular value of the given transfer function at its global maximizer on $\mathrm{i}\mathbb{R}$, and at nearby points. Hence, a value much smaller than the actual $\mathcal{L}_\infty$-norm is returned. Note that the peak at which the $\mathcal{L}_\infty$-norm is attained is so sharp that it was not captured with the original plotting resolution, so we had to manually add this maximizer. Thus, if the methods described above are not started at a point very close to the global maximizer, such sharp peaks are easily missed.

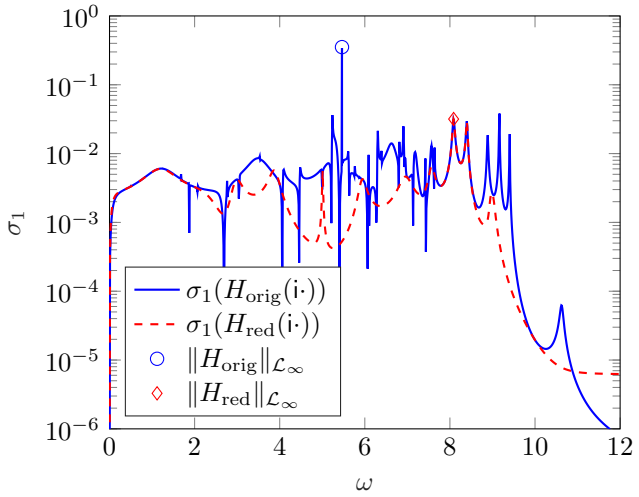


Figure 1. Maximum singular value plots of the original transfer function H_{orig} (depicted with a continuous blue line), and the final reduced transfer function H_{red} after applying `linorm_subsp` (plotted with red dashed line) with their respective $\mathcal{L}_\infty$-norms highlighted.

Therefore, it is necessary to certify the results of the algorithms for large-scale $\mathcal{L}_\infty$-norm computations, in particular to determine whether stagnation at a local maximizer that is not globally optimal has occurred. A requirement for certification methods is that they should not require a priori knowledge about the system to return the correct result.

Such a certification routine may be used in combination with iterative algorithms that minimize the $\mathcal{L}_\infty$-norm of a large-scale transfer function to optimize parameters or to synthesize robust controllers. In this case, the certification can be run either at the end of the minimization algorithm or every few iterations to check if, in fact, the correct $\mathcal{L}_\infty$-norm and not just a local maximizer of $\sigma_1(H(\mathrm{i}\cdot))$ is minimized. Hence, the certification does not need to be as fast as the $\mathcal{L}_\infty$-norm computation methods, but should still exploit sparsity of the system matrices.

1.2 Contributions and Outline

In this paper, we make the following main contributions.

- We develop an algorithm to determine whether a given value is lower than the $\mathcal{L}_\infty$-norm of a transfer function that does not require nontrivial user input.
- We implement [3] and heavily test the proposed method on a benchmark collection of large-scale systems to show its practical applicability.

The text is organized as follows. In the next section, we introduce the new algorithm for global optimality certification. We first show how to compute or estimate a critical interval in which $\sigma_1(H(\mathrm{i}\cdot))$ may be larger than some provided value γ. Then, we explain how this interval

[1] code available at `https://www.math.tu-berlin.de/index.php?id=186267`

[2] example system data available at `http://slicot.org/20-site/126-benchmark-examples-for-model-reduction`

[3] code available at `https://doi.org/10.5281/zenodo.3546221` alongside scripts to reproduce all figures in this paper

is scanned for values ω at which $\sigma_1(H(\mathrm{i}\omega)) = \gamma$. The technique used here is inspired by the complex frequency hopping introduced in Chiprout and Nakhla [1995], and employed in a similar way in Grivet-Talocia [2007] to enforce passivity of a given system. Another approach similar to ours is introduced in Gu et al. [2006], where the authors propose a method, based on shift-and-invert Arnoldi iterations, to find all real eigenvalues of a matrix in the context of computing the distance to uncontrollability of a linear control system. We also refer to Kressner [2006] concerning the use of similar ideas but for a special case of our setting, namely the distance to instability of a matrix.

The novelty of our work compared to Grivet-Talocia [2007] and Gu et al. [2006] is (beside the fact that this concerns the application of a similar scheme in a different context) that we pay special attention to the structure of the resulting eigenvalue problems during the shift-and-invert Arnoldi iterations. This allows to distinguish between eigenvalues that are purely imaginary and eigenvalues that have small nonzero real parts in absolute value in a reliable fashion. Furthermore, compared to Grivet-Talocia [2007], we do not require the a priori knowledge of all eigenvalues of the system matrix A, or the pencil $sE - A$. This makes the proposed method applicable to the large and sparse setting for which such information is only available, if at all, at an excessive computational cost.

Then, in Section 3, we conduct a series of numerical experiments in which we provide values for γ smaller and larger than the actual $\mathcal{L}_\infty$-norm of benchmark transfer functions. We check whether the algorithm returns the correct results in all such test cases. Finally, we outline other problems for which this algorithm may be used.

2. CERTIFICATION ALGORITHM

The algorithm is divided into two parts. First, as an initialization, an interval on the imaginary axis containing the points where the norm of the transfer function might be larger than the given value $\gamma > 0$ is determined. Secondly, this interval is scanned in finitely many iterations until either such a point is found or its existence is ruled out.

We make use of the fact that the pencil (2) (which we assume to be regular) has a purely imaginary eigenvalue if and only if $\gamma > \inf_{\omega\in\mathbb{R}} \|H(\mathrm{i}\omega)\|_2$ is less than or equal to the $\mathcal{L}_\infty$-norm of the given transfer function. Hence, our algorithm is designed to find imaginary eigenvalues of (2).

2.1 Initialization

We propose two different initialization techniques to determine an interval $\mathrm{i}[a, b] \subseteq \mathrm{i}\mathbb{R}$ in which imaginary eigenvalues of the pencil may occur. Since all matrices E, A, B, C, D are real, we may always choose $a = 0$ due to the spectral symmetry with respect to the real axis. Thus, it remains to find a suitable value for the upper bound b. For this, it suffices to compute the spectral radius

$$\rho_{\mathsf{fin}}(N, M_\gamma) := \max_{\lambda\in\mathbb{C}} \{|\lambda| \mid \det(\lambda N - M_\gamma) = 0\} \qquad (3)$$

of the finite part of the spectrum of the pencil $sN - M_\gamma$.

If N is singular, this upper bound cannot be as easily computed as the spectral radius. However, in some cases, we can transform the pencil $sN - M_\gamma$ into

$$\Pi_1^{\mathsf{T}} (sN - M_\gamma) \Pi_2 = s \begin{bmatrix} N_{11} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \qquad (4)$$

by some permutation matrices Π_1 and Π_2 such that N_{11}, M_{11}, M_{12}, M_{21}, M_{22} are still sparse, and N_{11} is invertible. If M_{22} is invertible, $\rho_{\mathsf{fin}}(N, M_\gamma)$ is equal to the spectral radius of $sN_{11} - (M_{11} - M_{12}M_{22}^{-1}M_{21})$, which can be computed by finding the eigenvalue with largest modulus using the Arnoldi iteration. The Arnoldi iteration has the tendency to converge to the outermost eigenvalues first.

Note that despite the fact that sparsity is lost in the new matrix $M_{11} - M_{12}M_{22}^{-1}M_{21}$, the Arnoldi iteration can still be applied efficiently without explicitly forming this matrix, as the Arnoldi iteration requires only the products of this matrix with vectors. Such a product can be evaluated efficiently by three matrix-vector multiplications, and one solve with M_{22}. In this way, the finite eigenvalue with largest modulus can be found efficiently, and thus the upper bound of the critical interval b.

For invertible E, the decomposition in (4) can be computed directly by using the Schur complement. Namely, if, additionally, $R := D^{\mathsf{T}}D - \gamma^2 I_m$ and $S := DD^{\mathsf{T}} - \gamma^2 I_p$ are invertible, then the finite eigenvalues of $sN - M_\gamma$ coincide with the eigenvalues of

$$s \begin{bmatrix} E & 0 \\ 0 & E^{\mathsf{T}} \end{bmatrix} - \begin{bmatrix} A - BR^{-1}D^{\mathsf{T}}C & -\gamma BR^{-1}B^{\mathsf{T}} \\ \gamma C^{\mathsf{T}}S^{-1}C & -A^{\mathsf{T}} + C^{\mathsf{T}}DR^{-1}B^{\mathsf{T}} \end{bmatrix}.$$

The previously discussed ways for computing $\rho_{\mathsf{fin}}(N, M_\gamma)$ are only efficiently applicable if N can be decomposed into the form (4) where the transformed matrices are sparse and M_{22} is invertible. This is not always the case. Especially, if the matrix pencil $sE - A$ has a higher index, i. e., the infinite eigenvalues have nontrivial Jordan blocks in the Weierstraß canonical form, then this may not be achieved. However, in many applications such as fluid dynamics or mechanical systems, the infinite eigenvalues of $sE - A$ can be implicitly deflated by applying certain projectors, see Mehrmann and Stykel [2005] for details.

As an alternative, we also propose a heuristic to estimate the critical interval as described in Alg. 1. This heuristic does require a specified tolerance ε, however, it is only used when the first option of computing the upper bound is not applicable. Furthermore, we have successfully tested our certification algorithm with the initialization provided in Alg. 1 on a diverse set of test examples, all with $\varepsilon = 10^{-6}$.

Algorithm 1 Heuristic to determine upper bound b

Input: Transfer function H, tolerance ε.
Output: Upper bound of the critical interval $b \in \mathbb{R}$.
1: Set $\omega := 1$.
2: Set $G := 0_{p\times m}$.
3: **while** $\|H(\mathrm{i}\omega) - G\|_2 > \varepsilon$ **do**
4: Set $G := H(\mathrm{i}\omega)$.
5: Set $\omega := 2\omega$.
6: **end while**
7: Set $b := \omega$.

At the end of the initialization stage based on either the decomposition (4) or the heuristic approach from Alg. 1, a finite interval on the imaginary axis, in which eigenvalues of the pencil in (2) may occur, is available.

Next, we explain how we determine whether eigenvalues of (2) indeed exist in this interval.

2.2 Main Routine

Before discussing the critical steps of the proposed method in detail, we first explain its main steps and how it functions. After initialization, we are given the critical interval $\mathrm{i}[0, b] \subset \mathrm{i}\mathbb{R}$, and we must check whether there are any eigenvalues of $sN - M_\gamma$ in this interval. For this, $\mathrm{i}\theta \in \mathrm{i}\mathbb{R}$ is set equal to the midpoint of the critical interval. A few eigenvalues of $sN - M_\gamma$ close to $\mathrm{i}\theta$ are computed using the Arnoldi iteration (see, e. g., Bai et al. [2000]) in conjunction with a spectral transformation.

If none of these eigenvalues is purely imaginary, then we consider the disk around $\mathrm{i}\theta$ of radius r, where r is the distance from $\mathrm{i}\theta$ to the eigenvalues closest to it. By construction, this disk does not contain any eigenvalues. Hence, the interval $\mathrm{i}(\theta - r, \theta + r)$ which is a subset of this disk cannot contain any purely imaginary eigenvalues.

As a result, we only need to look for purely imaginary eigenvalues in the set $\mathrm{i}[0, b] \setminus \mathrm{i}(\theta - r, \theta + r)$, which is either empty, leading to the termination of the algorithm, or the union of $\mathrm{i}[0, \theta - r]$ and $\mathrm{i}[\theta + r, b]$. In the latter case, in a recursive fashion, these two intervals can be checked for purely imaginary eigenvalues in the same way.

In Fig. 2, a visualization of this method is shown. There, the typical behavior of the algorithm can be observed. First, the original interval is divided into smaller intervals. These are then completely removed as the eigenvalue free disks start to intersect.

In Alg. 2, our method is described in detail. After initialization (lines 1–2), the matrix CI with its ijth entry denoted by CI_{ij}, and in which the bounds of the critical intervals are stored, is formed. The number of intervals n_{CI} that need to be checked is the number of rows of CI. In lines 5–7, an early stopping condition is checked. If the norm of the transfer function evaluated at the midpoint of the original interval is already larger than the specified value γ, there is no need to perform a more involved search for eigenvalues on the imaginary axis. In this case, the algorithm terminates before entering the main loop.

The main loop of the algorithm can be separated into two steps: an *eigenvalue check* (lines 9–19) and a *conclusion step* (lines 20–26). In each iteration, these two steps are performed to determine whether the algorithm terminates, and, if the algorithm does not terminate, the currently checked interval is either deleted or divided into two smaller intervals.

During the *eigenvalue check*, the shift $\mathrm{i}\theta$ is chosen as the midpoint of the currently considered interval; in Alg. 2, this interval is always defined by the first row in CI. Then, eigenvalues Λ close to this shift are computed by means of the Arnoldi iteration using a spectral transformation that is explained in Section 2.3. Finally, for every imaginary eigenvalue in Λ, we check whether the norm of the transfer function is in fact larger than γ around the imaginary eigenvalue by sampling the transfer function around this computed eigenvalue in an interval on the imaginary axis of length $2 \cdot \varepsilon_{\mathsf{eigs}}$. Here, $\varepsilon_{\mathsf{eigs}}$ is the convergence tolerance of the Arnoldi method. The necessity for this extra check is further explained in the next section.

In the *conclusion step*, first the radius r of the eigenvalue free disk around $\mathrm{i}\theta$ is computed as the minimum distance of $\mathrm{i}\theta$ to an eigenvalue in Λ. Then the bounds of the remaining subintervals are appended to the interval matrix CI if they are nonempty. Finally, the considered interval is removed from CI.

Algorithm 2 Certification routine to check $\|H\|_{\mathcal{L}_\infty} > \gamma$

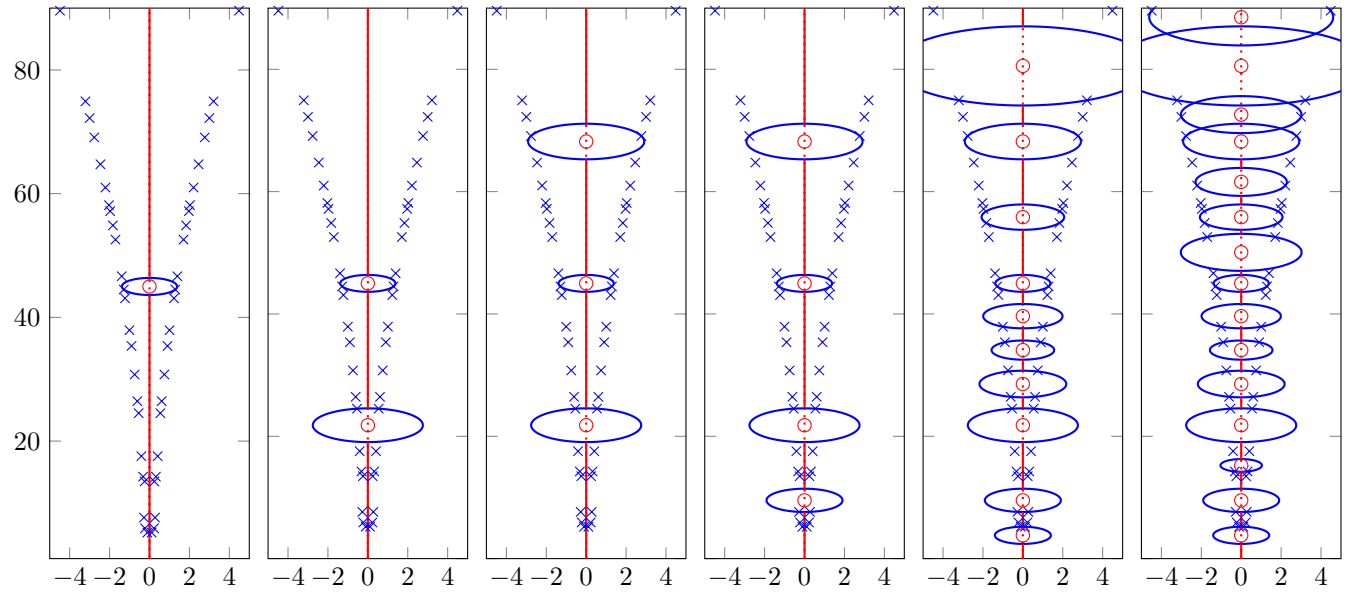
Input: Descriptor system matrices E, A, B, C, D defining (1), suspected $\mathcal{L}_\infty$-norm value $\gamma > 0$.
Output: Returns `true`, if $\|H\|_{\mathcal{L}_\infty} > \gamma$; `false` otherwise.

1: Construct even pencil $sN - M_\gamma$ as in (2).
2: Estimate upper bound b as in Section 2.1.
3: Set critical interval matrix $\mathsf{CI} := [0 \;\; b]$.
4: Set $n_{\mathsf{CI}} := 1$.
5: **if** $\|C(\mathrm{i}\frac{b}{2}E - A)^{-1}B + D\|_2 > \gamma$ **then**
6: Return `true`.
7: **end if**
8: **while** $n_{\mathsf{CI}} > 0$ **do**
9: Set $\theta := (\mathsf{CI}_{11} + \mathsf{CI}_{12})/2$.
10: Compute $\Lambda := \{\lambda_1, \ldots, \lambda_\ell\}$ = eigenvalues of $sN - M_\gamma$ close to $\mathrm{i}\theta$.
11: **for** λ_i with $\mathrm{Re}(\lambda_i) = 0$ **do**
12: Set $h = -\varepsilon_{\mathsf{eigs}}$.
13: **for** $k = 1, 2, 3$ **do**
14: **if** $\|C((\lambda_i + \mathrm{i}h)E - A)^{-1}B + D\|_2 > \gamma$ **then**
15: Return `true`.
16: **end if**
17: Set $h := h + \varepsilon_{\mathsf{eigs}}$.
18: **end for**
19: **end for**
20: Set $r := \min\{|\lambda_1 - \mathrm{i}\theta|, \ldots, |\lambda_\ell - \mathrm{i}\theta|\}$.
21: **if** $r < (\mathsf{CI}_{12} - \mathsf{CI}_{11})/2$ **then**
22: Set $\widetilde{\mathsf{CI}} := \begin{bmatrix} \mathsf{CI}_{11} & \theta - r \\ \theta + r & \mathsf{CI}_{12} \end{bmatrix}$.
23: Set $\mathsf{CI} := \begin{bmatrix} \mathsf{CI} \\ \widetilde{\mathsf{CI}} \end{bmatrix}$.
24: Set $n_{\mathsf{CI}} := n_{\mathsf{CI}} + 2$.
25: **end if**
26: Delete first row of CI and set $n_{\mathsf{CI}} := n_{\mathsf{CI}} - 1$.
27: **end while**
28: Return `false`.

2.3 Spectral Transformation and Eigenvalue Computation

The spectrum of a real, even pencil such as (2) is symmetric with respect to both the imaginary and real axis, such that for an eigenvalue λ_i of (2), we have that $-\lambda_i$, $\overline{\lambda}_i$, and $-\overline{\lambda}_i$ are also eigenvalues of (2). This symmetry can be be exploited during eigenvalue computation to distinguish imaginary eigenvalues from eigenvalues that are just close to the imaginary axis. As the identification of imaginary eigenvalues is an important part of our method, symmetry preservation is essential.

Since with the Arnoldi iteration the eigenvalues largest in modulus are computed, a spectral transformation must be applied when the eigenvalues closest to some shift ξ are desired. The standard transformation used is the *shift-and-invert transformation*, which results in the pencil



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4 (e) Iteration 10 (f) Iteration 15

Figure 2. Iterations of Alg. 2 for the `build` system (available in the SLICOT benchmark collection for model reduction) with $\gamma = 5.803963 \cdot 10^{-3}$. The eigenvalues of $sN - M_\gamma$ and the shifts are depicted with blue crosses and red circles, respectively, the critical intervals are shown as red lines, and the blue circles represent the eigenvalue free regions.

$s(M_\gamma - \xi N) - N$ with the eigenvalue $\eta_i := (\lambda_i - \xi)^{-1}$, corresponding to an eigenvalue λ_i of (2).

This transformation, however, does not preserve the spectral symmetry of (2). As a result, the imaginary eigenvalues of the original pencil are usually approximated by eigenvalues that are just close to the imaginary axis. This causes difficulties, since in Alg. 2 imaginary eigenvalues need to be distinguished from eigenvalues that have nonzero real parts small in absolute value.

Instead, we use the symmetry-preserving transformation from Mehrmann and Watkins [2001] and from Mehrmann et al. [2012] for singular N, resulting in the matrix eigenvalue problem $(\eta I - K(\xi))x = 0$ for $x \neq 0$, with

$$K(\xi) := (M_\gamma + \xi N)^{-1} N (M_\gamma - \xi N)^{-1} N. \tag{5}$$

It can be verified that if λ_i is an eigenvalue of $sN - M_\gamma$, then $\eta_i = 1/(\lambda_i^2 - \xi^2)$ is an eigenvalue of $K(\xi)$. This implies that the eigenvalues λ_i and $-\lambda_i$ of (2) are mapped onto the same eigenvalue of $K(\xi)$. Note that also the spectral symmetry with respect to the real axis is preserved by this transformation for imaginary or real ξ, since it is shown in Mehrmann et al. [2012] that then $K(\xi)$ is a real matrix.

The benefit of using the transformation (5) over the classical shift-and-invert strategy is that computed eigenvalues that are originally on the imaginary axis have a zero real part exactly when the transformation (5) is used in Alg. 2, so we do not need to set a somewhat arbitrary tolerance to distinguish imaginary eigenvalues from the rest of the spectrum. On the other hand, for the eigenvalues computed by the classical shift-and-invert Arnoldi, the spectral symmetry is lost and the computed eigenvalues are not located precisely on the imaginary axis.

One problem that must still be addressed is the occurrence of an eigenvalue pair close to the imaginary axis which may be computed as an imaginary eigenvalue pair due to numerical errors. This can happen despite symmetry preservation because the symmetry is not broken if both eigenvalues in that pair become purely imaginary. For our method, this implies that even though $\gamma > \|H\|_{\mathcal{L}_\infty}$, it may happen that in line 10 of Alg. 2 eigenvalues with zero real part are returned. Therefore, an extra check is performed in lines 11–19, in which the transfer function is evaluated around the computed imaginary eigenvalues to check whether the transfer function evaluated near the imaginary eigenvalue has in fact a norm larger than γ.

3. NUMERICAL EXPERIMENTS

The test setup for testing our method is straightforward. We test the method on a set of 33 benchmark examples taken from Chahlaoui and Van Dooren [2002], Rommes and Martins [2006], Martins et al. [2007], Freitas et al. [2008] that are regularly used to test $\mathcal{L}_\infty$-norm computations and model order reduction methods. Furthermore, we choose a factor α_j from the set of values $\{\alpha_1 \ldots, \alpha_8\} := \{10^{-5}, 10^{-3}, 10^{-1}, 0.99, 1.01, 1.1, 2, 10\}$.

Then, for all benchmark systems with system matrices E_i, A_i, B_i, C_i, D_i and transfer function H_i, we pass E_i, A_i, B_i, C_i, and D_i, as well as a value $\gamma_j := \alpha_j \cdot \|H_i\|_{\mathcal{L}_\infty}$ for $i = 1, \ldots, 33$ and $j = 1, \ldots, 8$ to our implementation of Alg. 2. In this way, we obtain 264 test examples. The verification condition is that Alg. 2 must return `true` for $\alpha_j < 1$ and `false` for $\alpha_j > 1$.

Our algorithm returns the correct results for all 264 test cases. The tests are available with the MATLAB implementation of our method. In our implementation, we remove intervals of length smaller than $\varepsilon_{\mathsf{eigs}}$ from CI. These small intervals only occur due to errors in the eigenvalue computation, and are often close to eigenvalue

Table 1. Cumulative runtimes of Alg. 1 and 2

		runtime in seconds		
system	dimension q	α_3	α_4	α_5
`build`	98	0.09	0.18	0.27
`peec`	962	1.79	8.21	37.39
`M80PI_n1`	8062	94.16	286.39	659.82
`M80PI_n`	8370	92.90	353.23	653.40
`bips98_1450`	22618	11.19	37.15	115.37
`xingo_afonso_itaipu`	26502	25.62	33.80	89.31
`mimo8x8_system`	26634	26.28	35.98	102.67
`mimo46x46_system`	26592	41.18	41.10	110.12
`bips07_3078`	42264	10.99	37.92	95.17

pairs with small nonzero real parts. If a shift is placed close to such a pair, the matrix $K(\xi)$ is ill-conditioned and `eigs` may fail (since the desired accuracy cannot be achieved). Without the removal of these small intervals, our algorithm terminates with an exception during the execution of the MATLAB function `eigs` for 7 out of 264 examples. With the removal enabled, all test examples pass. We report the runtimes for a few examples in Table 1. All examples have been solved in several minutes at most. The runtime does not scale with the problem dimension, but is highly dependent on the particular problem.

4. CONCLUSION AND OUTLOOK

In this work, we have developed and implemented a new method for determining whether a large and sparse even matrix pencil has imaginary eigenvalues. We have illustrated that the method facilitates the reliable computation of the $\mathcal{L}_\infty$-norm in the large-scale setting. Further applications include the verification of passivity or contractivity of linear control systems, as well as checking feasibility of linear-quadratic regular problems, see Benner et al. [2015].

REFERENCES

N. Aliyev, P. Benner, E. Mengi, P. Schwerdtner, and M. Voigt. Large-scale computation of $\mathcal{L}_\infty$-norms by a greedy subspace method. *SIAM J. Matrix Anal. Appl.*, 38(4):1496–1516, 2017.

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems*. SIAM, 2000. ISBN 978-0-89871-471-5.

P. Benner, V. Sima, and M. Voigt. $\mathcal{L}_\infty$-norm computation for continuous-time descriptor systems using structured matrix pencils. *IEEE Trans. Automat. Control*, 57(1): 233–238, 2012.

P. Benner, P. Losse, V. Mehrmann, and M. Voigt. Numerical linear algebra methods for linear differential-algebraic equations. In A. Ilchmann and T. Reis, editors, *Surveys in Differential-Algebraic Equations III*, Differ.-Algebr. Equ. Forum, chapter 3, pages 117–175. Springer-Verlag, Cham, Switzerland, 2015. ISBN 978-3-319-22427-5.

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(3):24:1–24:26, 2016.

S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. *Systems Control Lett.*, 15(1):1–7, 1990.

N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function matrix. *Systems Control Lett.*, 14(4):287–293, 1990.

Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. SLICOT Working Note 2002–2, February 2002.

E. Chiprout and M. S. Nakhla. Analysis of interconnect networks using complex frequency hopping (CFH). *IEEE Trans. Computer-Aided Design*, 14(2):186–200, 1995.

M. A. Freitag, A. Spence, and P. Van Dooren. Calculating the H_∞-norm using the implicit determinant method. *SIAM J. Matrix Anal. Appl.*, 35(2):619–635, 2014.

F. Freitas, J. Rommes, and N. Martins. Gramian-based reduction method applied to large sparse power system descriptor models. *IEEE Trans. Power Syst.*, 23(3): 1258–1270, 2008.

S. Grivet-Talocia. An adaptive sampling technique for passivity characterization and enforcement of large interconnect macromodels. *IEEE Trans. Adv. Packag.*, 30 (2):226–237, 2007.

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

N. Guglielmi, M. Gürbüzbalaban, and M. L. Overton. Fast approximation of the H_∞-norm via optimization over spectral value sets. *SIAM J. Matrix Anal. Appl.*, 34(2): 709–737, 2013.

D. Kressner. Finding the distance to instability of a large sparse matrix. In *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 31–36, Munich, Germany, October 2006.

N. Martins, P. C. Pellanda, and J. Rommes. Computation of transfer function dominant zeros with applications to oscillation damping control of large power systems. *IEEE Trans. Power Syst.*, 22(4):1657–1664, 2007.

V. Mehrmann and T. Stykel. Balanced truncation model reduction for large-scale systems in descriptor form. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes Comput. Sci. Eng.*, chapter 3, pages 89–116. Springer-Verlag, Berlin, Heidelberg, New York, 2005. ISBN 978-3-540-24545-2.

V. Mehrmann and D. Watkins. Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils. *SIAM J. Sci. Comput.*, 22(6):1905–1925, 2001.

V. Mehrmann, C. Schröder, and V. Simoncini. An implicitly-restarted Krylov subspace method for real symmetric/skew-symmetric eigenproblems. *Linear Algebra Appl.*, 436(10):4070–4087, 2012.

J. Rommes and N. Martins. Efficient computation of transfer function dominant poles using subspace acceleration. *IEEE Trans. Power Syst.*, 21(3):1218–1226, 2006.

P. Schwerdtner and M. Voigt. Computation of the $\mathcal{L}_\infty$-norm using rational interpolation. *IFAC-PapersOnLine*, 51(25):84–89, 2018. 9th IFAC Symposium on Robust Control Design 2018, Florianópolis, Brazil.

M. Voigt. *On Linear-Quadratic Optimal Control and Robustness of Differential-Algebraic Systems*. Logos-Verlag, Berlin, 2015. ISBN 978-3-8325-4118-7.