SIAM J. MATRIX ANAL. APPL.
Vol. 35, No. 2, pp. 619–635

CALCULATING THE H_∞-NORM USING THE IMPLICIT DETERMINANT METHOD*

MELINA A. FREITAG†, ALASTAIR SPENCE‡, AND PAUL VAN DOOREN§

Abstract. We propose a fast algorithm to calculate the H_∞-norm of a transfer matrix. The method builds on a well-known relationship between singular values of the transfer function and pure imaginary eigenvalues of a certain Hamiltonian matrix. Using this property we construct a two-parameter eigenvalue problem, where, in the generic case, the critical value corresponds to a two-dimensional Jordan block. We use the implicit determinant method which replaces the need for eigensolves by the solution of linear systems, a technique recently used in [M. A. Freitag and A. Spence, *Linear Algebra Appl.*, 435 (2011), pp. 3189–3205] for finding the distance to instability. In this paper the method takes advantage of the structured linear systems that arise within the algorithm to obtain efficient solves using the staircase reduction. We give numerical examples and compare our method to the algorithm proposed in [N. Guglielmi, M. Gürbüzbalaban, and M. L. Overton, *SIAM J. Matrix Anal. Appl.*, 34 (2013), pp. 709–737].

Key words. H_∞-norm, Hamiltonian matrix, transfer matrix, Newton method

AMS subject classifications. 15A18, 93C80, 65F15

DOI. 10.1137/130933228

1. Introduction.

The H_∞-norm of a transfer function matrix is an important property for measuring robust stability in classical control theory [4]. In this paper we introduce a new fast method for calculating this quantity by extending an algorithm recently introduced in [13] (see also [2, 1, 28]).

Consider the continuous time linear dynamical system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1.1}$$

where $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$, $C \in \mathbb{C}^{m\times n}$, and $D \in \mathbb{C}^{m\times p}$. We will focus on continuous time systems, but the idea and method extends to discrete time problems (as was shown for the distance to instability in the discrete case in [21, Chapter 3]). Moreover, the method extends to linear time-invariant descriptor systems.

It is standard to call $G(s) = C(sI - A)^{-1}B + D$ the transfer matrix of the system (1.1). Let A be stable, that is, let all the eigenvalues of A have negative real part. Then the H_∞-norm of the transfer matrix $G(s)$ is defined as the supremum of the maximum singular value of the transfer function $G(s)$ evaluated on the imaginary axis, that is,

$$\|G\|_\infty := \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(i\omega)), \tag{1.2}$$

*Received by the editors August 16, 2013; accepted for publication (in revised form) by W.-W. Lin February 7, 2014; published electronically May 15, 2014.
http://www.siam.org/journals/simax/35-2/93322.html

†Corresponding author. Department of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, UK (m.freitag@maths.bath.ac.uk).

‡Department of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, UK (a.spence@bath.ac.uk).

§Department of Mathematical Engineering, Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (paul.vandooren@uclouvain.be). This author was partially supported by the Belgian Network DYSCO, funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

where $\sigma_{\max}$ denotes the maximum singular value of the matrix. Note that for $B = C = I$ and $D = 0$ the reciprocal of $\|G\|_\infty$ is referred to as the distance to instability [30, 10]. The H_∞-norm is used in several applications, for example, in robust control (e.g., [12]) or as an error measure for model order reduction (e.g., [18]). In this paper we introduce a new quadratically convergent method of estimating the H_∞-norm for the continuous time linear dynamical system given by (1.1). This method builds on two foundations. First, we use the relation between the singular values of a transfer function matrix and the eigenvalues of a certain Hamiltonian matrix introduced in [10] and exploited in [7]. Second, we use the implicit determinant method which has its roots in bifurcation analysis (see [17]) and has recently been applied to the problem of finding the stability radius of continuous linear systems [13] (and later extended to discrete systems [21, Chapter 3]).

The standard and most well-known method to compute the H_∞-norm is the Boyd–Balakrishnan–Bruinsma–Steinbuch algorithm [6, 8] which requires repeated computation of all the eigenvalues of a Hamiltonian matrix. A new method to compute the H_∞-norm was recently suggested in [19] and [5]. Both approaches are generalizations of the method in [20]. The former uses spectral value sets as generalizations to the matrix pseudospectrum and fast approximations to the spectral value set abscissa, while the latter uses structured pseudospectra to extend the idea to descriptor systems.

The plan of this paper is as follows. In section 2 the optimization problem (1.2) is reformulated to one of finding zeros of the determinant of a parameter-dependent Hermitian matrix. Next, in section 3, the implicit determinant method (see [28]) is described and applied to the problem of this paper. The implementation of the implicit determinant method is discussed in section 4, and numerical examples that illustrate the performance of the method are given in section 5.

2. Reformulation of the problem. In this section the problem of finding the supremum over ω of $\sigma_{\max}(G(i\omega))$, as in (1.2), is reformulated to one of finding zeros of the determinant of a parameter-dependent Hermitian matrix. This formulation is amendable to attack using the implicit determinant method as in [13]. Several intermediate steps as well as an important assumption are required for this reformulation as we now describe.

First, recall the following equivalence between singular values of the transfer function and imaginary eigenvalues of a related Hamiltonian matrix described in [7] (see also [6, 8]).

THEOREM 2.1. *Consider the system* (1.1) *with transfer function matrix* $G(s) = C(sI - A)^{-1}B + D$. *Let* A *be stable and assume* $\gamma > \sigma_{\max}(D)$. *Define the matrix* $M(\gamma)$ *by*

$$M(\gamma) = \begin{bmatrix} A - BR(\gamma)^{-1}D^H C & -\gamma BR(\gamma)^{-1}B^H \\ \gamma C^H S(\gamma)^{-1}C & -A^H + C^H D R(\gamma)^{-1}B^H \end{bmatrix}, \tag{2.1}$$

where $R(\gamma) = D^H D - \gamma^2 I$ *and* $S(\gamma) = DD^H - \gamma^2 I$. *Then* $\|G\|_\infty \geq \gamma$ *if and only if the matrix* $M(\gamma)$ *has pure imaginary eigenvalues.*

Proof. See [7] for the proof. ∎

It is shown in [7] that γ is a singular value of $G(i\omega)$ if and only if $M(\gamma) - i\omega I$ is singular, where $\omega \in \mathbb{R}$. Theorem 2.1 says we need to find the largest possible value of γ such that $M(\gamma)$ has imaginary eigenvalues. If we denote this largest possible critical value of γ by γ^*, then we have $\gamma^* = \|G\|_\infty$.

The matrix $M(\gamma)$ is Hamiltonian, that is, $(JM(\gamma))^H = JM(\gamma)$, where $J = \left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$. One can also show by similarity transform that $M(\gamma)$ has the same eigenvalues as the Hamiltonian matrix $H(\gamma)$ given by

$$H(\gamma) = \begin{bmatrix} A & 0 \\ C^H C & -A^H \end{bmatrix} + \begin{bmatrix} B \\ C^H D \end{bmatrix} R(\gamma)^{-1} \begin{bmatrix} -D^H C & B^H \end{bmatrix}. \tag{2.2}$$

The eigenvalues of $H(\gamma)$ are symmetric about the imaginary axis, that is, if (λ, x) is a right eigenpair of $H(\gamma)$, then $(-\bar{\lambda}, (Jx)^H)$ is a left eigenpair of $H(\gamma)$. Note that if $H(\gamma^*)$ is real, then the eigenvalues of $H(\gamma^*)$ are also symmetric about the real axis.

Due to the Hamiltonian structure of $H(\gamma)$ every pure imaginary eigenvalue $i\omega$ of $H(\gamma)$ has even algebraic multiplicity at the critical point (γ^*, ω^*). We make the following assumption on the critical pure imaginary eigenvalue $i\omega^*$ of $H(\gamma^*)$ (corresponding to the critical value $\gamma^* = \|G\|_\infty$).

ASSUMPTION 2.2. *$(i\omega^*, x^*)$ is a defective eigenpair of $H(\gamma^*)$ of algebraic multiplicity* 2.

Throughout this paper we shall assume that Assumption 2.2 holds. It can be shown that if a singular value γ of $G(i\omega)$ is simple, then the pure imaginary eigenvalue $i\omega$ of $H(\gamma)$ is defective, or, $\ker(H(\gamma) - i\omega I) = 1$. This is actually a necessary and sufficient condition (see, for example, [14]). If we assume the largest singular value γ^* of $G(i\omega^*)$ is simple, then $(i\omega^*, x^*)$ is a defective eigenpair of $H(\gamma^*)$. This assumption is also made in [19] and generically the largest singular value of $G(i\omega)$ is simple (see [19, 9]). With the property that every pure imaginary eigenvalue $i\omega$ of the Hamiltonian matrix $H(\gamma)$ has even algebraic multiplicity at the critical point (γ^*, ω^*) we obtain that Assumption 2.2 is generically true. Of course, there is the possibility of a larger (even) algebraic multiplicity of a Jordan block. However, our algorithm will detect these cases as we shall show.

Under Assumption 2.2 we have that the eigenvalue $i\omega^*$ appears in a Jordan block of dimension 2, that is

$$(H(\gamma^*) - i\omega^* I)x^* = 0, \quad x^* \neq 0, \quad \text{and} \quad \dim\ker(H(\gamma^*) - i\omega^* I) = 1, \tag{2.3}$$

and if we denote the left eigenvector of $H(\gamma^*) - i\omega^* I$ by y^*, then ${y^*}^H x^* = 0$. Using the fact that $y^* = Jx^*$ as well as $J^{-1} = J^H = -J$ leads to

$${x^*}^H J x^* = 0, \tag{2.4}$$

which is a direct consequence of Assumption 2.2. Furthermore, if $\hat{x}^*$ is the generalized eigenvector of $i\omega^*$, then under Assumption 2.2 (that is, a Jordan block of dimension 2 at the critical value) we have

$$(H(\gamma^*) - i\omega^* I)\hat{x}^* = x^* \quad \text{and} \quad {y^*}^H \hat{x}^* \neq 0. \tag{2.5}$$

We remark that for higher dimensional Jordan blocks we would have ${y^*}^H \hat{x}^* = 0$.

Note that if the property (2.4) is satisfied, then the eigenvector x^* is called J-neutral (see [3, p. 487]) and the corresponding pure imaginary eigenvalue $i\omega^*$ is said to have mixed sign characteristic (see [3, Proposition 4.10]).

We next proceed by converting the Hamiltonian eigenvalue problem for $H(\gamma)$ in (2.2) into a parameter-dependent *Hermitian eigenproblem that avoids computing the inverse of* $R(\gamma)$. Multiplying $H(\gamma) - i\omega I$ in (2.2) by J from the left yields a Hermitian

eigenvalue problem where $\hat{H}(\gamma) := H(\gamma)J$ and for the critical values γ^* and ω^* we have (cf. (2.3))

$$(\hat{H}(\gamma^*) - i\omega^* J)Jx^* = 0, \tag{2.6}$$

where $y^* = Jx^*$ is now both the right and left eigenvector and the Hermitian matrix $\hat{H}(\gamma^*)$ given by

$$\hat{H}(\gamma^*) = \begin{bmatrix} 0 & A \\ A^H & C^H C \end{bmatrix} - \begin{bmatrix} B \\ C^H D \end{bmatrix} R(\gamma^*)^{-1} \begin{bmatrix} B^H & D^H C \end{bmatrix}$$

(see also [15]). The pencil $\hat{H}(\gamma^*) - i\omega^* J$ is the Schur complement of

$$\mathcal{H}(\gamma^*, \omega^*) = \begin{bmatrix} 0 & A - i\omega^* I & B \\ A^H + i\omega^* I & C^H C & C^H D \\ B^H & D^H C & D^H D - {\gamma^*}^2 I \end{bmatrix} \in \mathbb{C}^{2n+p, 2n+p}, \tag{2.7}$$

and hence, if γ^* is not a singular value of D, which is satisfied under the conditions of Theorem 2.1, then the following equivalence holds:

$$\det \mathcal{H}(\gamma^*, \omega^*) = 0 \quad \Leftrightarrow \quad \det(\hat{H}(\gamma^*) - i\omega^* J) = 0.$$

Our reformulation is almost complete: The singular value problem (1.2) has been reformulated as finding (γ^*, ω^*) such that

$$\det \mathcal{H}(\gamma^*, \omega^*) = 0.$$

The last step in this process is to border the matrix $\mathcal{H}(\gamma^*, \omega^*)$ in such a way that Theorem 2.4 holds. Before stating this theorem we recall Lemma 2.8 of [23] (see also [27, Lemma 3.3]), stated here for Hermitian problems, as it is used to prove Theorem 2.4.

LEMMA 2.3. *Given an $r \times r$ Hermitian matrix K with rank$(K) = r - 1$, and $w \in \mathbb{C}^r$. The $(r+1) \times (r+1)$ Hermitian bordered matrix*

$$L = \begin{bmatrix} K & w \\ w^T & 0 \end{bmatrix}$$

is nonsingular if and only if $w^T \Phi \neq 0$ for all $\Phi \in ker(K) \setminus \{0\}$.

THEOREM 2.4. *Let Assumption 2.2 be satisfied, assume γ^* is not a singular value of D and the vector $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right] \in \mathbb{C}^{2n}$, $v_1, v_2 \in \mathbb{C}^n$ satisfies*

$$v^H J x^* \neq 0, \quad \text{where} \quad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}. \tag{2.8}$$

Then the matrix $T(\gamma, \omega) \in \mathbb{C}^{2n+p+1 \times 2n+p+1}$ given by

$$T(\gamma, \omega) := \begin{bmatrix} & & & v_1 \\ & \mathcal{H}(\gamma, \omega) & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & A - i\omega I & B & v_1 \\ A^H + i\omega I & C^H C & C^H D & v_2 \\ B^H & D^H C & D^H D - \gamma^2 I & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \tag{2.9}$$

is nonsingular at $(\gamma, \omega) = (\gamma^*, \omega^*)$.

Proof. Using (2.3) and (2.6) we have that $\dim\ker(\hat{H}(\gamma^*) - i\omega^* J) = 1$. Furthermore, it is straightforward to check that Jx^* is an eigenvector of $\hat{H}(\gamma^*) - i\omega^* J$ (that is, (2.6) holds) if and only if

$$\mathcal{H}(\gamma^*, \omega^*)z^* = 0, \quad \text{where} \quad z^* = \begin{bmatrix} Jx^* \\ -(D^H D - \gamma^{*2} I)^{-1}[B^H \; D^H C]Jx^* \end{bmatrix}, \tag{2.10}$$

and hence $\dim\ker(\mathcal{H}(\gamma^*, \omega^*)) = 1$ with $z^* \in \ker(\mathcal{H}(\gamma^*, \omega^*))$. Lemma 2.3 with (2.10) then shows that $T(\gamma^*, \omega^*)$ is nonsingular if (2.8) holds. ∎

3. The implicit determinant method. In this section we use the result of Theorem 2.4 above to describe the implicit determinant method, a method designed to find values of γ and ω that satisfy $\det(\mathcal{H}(\gamma, \omega)) = 0$, and hence, calculate $\gamma^* = \|G\|_\infty$, the required H_∞-norm.

Consider $T(\gamma, \omega)$, which is nonsingular for (γ, ω) close to (γ^*, ω^*) (by [16, Theorem 2.3.4]). Then, near (γ^*, ω^*) the linear system

$$\begin{bmatrix} & \mathcal{H}(\gamma,\omega) & & v_1 \\ & & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1(\gamma,\omega) \\ z_2(\gamma,\omega) \\ z_3(\gamma,\omega) \\ f(\gamma,\omega) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.1}$$

where $z_1(\gamma, \omega), z_2(\gamma, \omega) \in \mathbb{C}^n$, $z_3(\gamma, \omega) \in \mathbb{C}^p$, and

$$\mathcal{H}(\gamma, \omega) = \begin{bmatrix} 0 & A - i\omega I & B \\ A^H + i\omega I & C^H C & C^H D \\ B^H & D^H C & D^H D - \gamma^2 I \end{bmatrix}$$

has a unique solution $[z_1 \; z_2 \; z_3 \; f]^H$, which is a $\mathcal{C}^\infty$-function of (γ, ω). The reason we consider the linear system (3.1) is that, by application of Cramer's rule, we have that

$$f(\gamma, \omega) = \frac{\det(\mathcal{H}(\gamma, \omega))}{\det(T(\gamma, \omega))},$$

and hence, as $T(\gamma, \omega)$ is nonsingular in a neighborhood of (γ^*, ω^*) we have that

$$f(\gamma, \omega) = 0 \quad \Leftrightarrow \quad \det(\mathcal{H}(\gamma, \omega)) = 0. \tag{3.2}$$

Note also that if $f(\gamma, \omega) = 0$, we have that $z \in \ker(\mathcal{H}(\gamma, \omega))$, where $z = [z_1 \; z_2 \; z_3]^H$. It is straightforward to show that $f(\gamma, \omega)$ is real since $\mathcal{H}(\gamma, \omega)$ is Hermitian.

The equivalence between zeros of $f(\gamma, \omega)$ and those of $\det(\mathcal{H}(\gamma, \omega))$ has several consequences. In practical terms, in the (γ, ω)-plane, solutions of $\det(\mathcal{H}(\gamma, \omega)) = 0$ may be computed by computing solutions of $f(\gamma, \omega) = 0$. In fact, there are paths of solutions of $\det(\mathcal{H}(\gamma, \omega)) = 0$ and these could be calculated by computing paths of $f(\gamma, \omega) = 0$ (though we do not do so in this paper). One extremely important property is that information about derivatives of $\det(\mathcal{H}(\gamma, \omega)) = 0$ is inherited by derivatives of $f(\gamma, \omega) = 0$, and through this relationship information about the Jordan structure in $\mathcal{H}(\gamma, \omega)$ may be extracted from derivatives of $f(\gamma, \omega) = 0$. This is explained in detail in [2]. We continue this theme by now showing how Assumption 2.2 and Theorems 2.1 and 2.4 impact on a solution curve $f(\gamma, \omega) = 0$.

We have the following lemma.

LEMMA 3.1. *Let the assumptions of Theorems* 2.1 *and* 2.4 *be satisfied, and consider* $f(\gamma, \omega)$ *to be one of the components in the solution to the linear system* (3.1).

Consider the curve $f(\gamma,\omega)=0$. Then, near (γ^*,ω^*) we have

$$f_\gamma(\gamma,\omega) = 2\gamma\|z_3(\gamma,\omega)\|^2 > 0. \tag{3.3}$$

Furthermore,

$$\text{(a)} \quad f_\omega(\gamma^*,\omega^*) = 0, \tag{3.4}$$

$$\text{(b)} \quad f_{\omega\omega}(\gamma^*,\omega^*) > 0. \tag{3.5}$$

Proof. Differentiating the linear system (3.1) with respect to γ gives

$$\begin{bmatrix} & \mathcal{H}(\gamma,\omega) & & v_1 \\ & & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \begin{bmatrix} z_\gamma(\gamma,\omega) \\ f_\gamma(\gamma,\omega) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2\gamma z_3(\gamma,\omega) \\ 0 \end{bmatrix}. \tag{3.6}$$

Consider the first block row

$$\mathcal{H}(\gamma,\omega)z_\gamma(\gamma,\omega) + f_\gamma(\gamma,\omega)\begin{bmatrix} v_1 \\ v_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2\gamma z_3(\gamma,\omega) \end{bmatrix}.$$

Multiplying this equation from the left by $z(\gamma,\omega)^H$, from the left null-vector of $\mathcal{H}(\gamma,\omega)$ we get

$$f_\gamma(\gamma,\omega)z(\gamma,\omega)^H \begin{bmatrix} v_1 \\ v_2 \\ 0 \end{bmatrix} = 2\gamma\|z_3(\gamma,\omega)\|^2,$$

and with the second row of (3.1) we obtain

$$f_\gamma(\gamma,\omega) = 2\gamma\|z_3(\gamma,\omega)\|^2 > 0,$$

which gives (3.3). The inequality follows from $\gamma > \sigma_{\max}(D)$ in the assumption of Theorem 2.1 and $z_3(\gamma,\omega) \neq 0$. (If $z_3(\gamma,\omega) = 0$, then (3.1) with $f = 0$ yields that A has eigenvalues on the imaginary axis, resulting in a contradiction to A being stable.)

Similarly, differentiate (3.1) with respect to ω and get

$$\begin{bmatrix} & \mathcal{H}(\gamma,\omega) & & v_1 \\ & & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \begin{bmatrix} z_\omega(\gamma,\omega) \\ f_\omega(\gamma,\omega) \end{bmatrix} = i \begin{bmatrix} J\begin{bmatrix} z_1(\gamma,\omega) \\ z_2(\gamma,\omega) \end{bmatrix} \\ 0 \\ 0 \end{bmatrix}. \tag{3.7}$$

Evaluate the first block row at the root (γ^*,ω^*) to obtain

$$\mathcal{H}(\gamma^*,\omega^*)z_\omega(\gamma^*,\omega^*) + f_\omega(\gamma^*,\omega^*)\begin{bmatrix} v_1 \\ v_2 \\ 0 \end{bmatrix} = i \begin{bmatrix} J\begin{bmatrix} z_1(\gamma^*,\omega^*) \\ z_2(\gamma^*,\omega^*) \end{bmatrix} \\ 0 \end{bmatrix}.$$

Multiply this equation by the left null-vector $z(\gamma^*,\omega^*)^H =: z^{*H}$ of $\mathcal{H}(\gamma^*,\omega^*)$ and use the normalization $z^{*H}\left[\begin{smallmatrix} v_1 \\ v_2 \\ 0 \end{smallmatrix}\right] = 1$ from the second row of (3.1) to get

$$f_\omega(\gamma^*,\omega^*) = i \begin{bmatrix} z_1(\gamma^*,\omega^*) & z_2(\gamma^*,\omega^*) \end{bmatrix}^H J \begin{bmatrix} z_1(\gamma^*,\omega^*) \\ z_2(\gamma^*,\omega^*) \end{bmatrix},$$

where we have used (2.8). Note that $z^* \in \ker(\mathcal{H}(\gamma^*, \omega^*))$ and z^* is given by (2.10), hence $\left[\begin{smallmatrix} z_1(\gamma^*,\omega^*) \\ z_2(\gamma^*,\omega^*) \end{smallmatrix}\right] = Jx^*$ and with $J^H = J^{-1}$ we have that

$$f_\omega(\gamma^*, \omega^*) = i{x^*}^H J x^* = 0,$$

where the last equality follows from (2.4) and hence (3.4) holds. Also, with (2.10) and (3.4) the first row of (3.7) evaluated at (γ^*, ω^*) leads to

$$\mathcal{H}(\gamma^*, \omega^*) z_\omega(\gamma^*, \omega^*) = -i \begin{bmatrix} x^* \\ 0 \end{bmatrix}.$$

Using the definition of $\mathcal{H}$ in (2.7) as well as (2.6) we obtain

$$(\hat{H}(\gamma^*) - i\omega^* J) \begin{bmatrix} (z_1)_\omega(\gamma^*, \omega^*) \\ (z_2)_\omega(\gamma^*, \omega^*) \end{bmatrix} = -ix^*,$$

and with (2.5) we have

$$\hat{x}^* = iJ \begin{bmatrix} (z_1)_\omega(\gamma^*, \omega^*) \\ (z_2)_\omega(\gamma^*, \omega^*) \end{bmatrix}. \tag{3.8}$$

Differentiating the linear system (3.7) with respect to ω again we get

$$\begin{bmatrix} & \mathcal{H}(\gamma,\omega) & & v_1 \\ & & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{\omega\omega}(\gamma,\omega) \\ f_{\omega\omega}(\gamma,\omega) \end{bmatrix} = 2i \begin{bmatrix} J \begin{bmatrix} (z_1)_\omega(\gamma,\omega) \\ (z_2)_\omega(\gamma,\omega) \end{bmatrix} \\ 0 \\ 0 \end{bmatrix}. \tag{3.9}$$

Evaluate at (γ^*, ω^*) and multiply the first block row by the left eigenvector ${z^*}^H$ to get

$$f_{\omega\omega}(\gamma^*, \omega^*) = 2i \begin{bmatrix} z_1(\gamma^*, \omega^*) & z_2(\gamma^*, \omega^*) \end{bmatrix} J \begin{bmatrix} (z_1)_\omega(\gamma^*, \omega^*) \\ (z_2)_\omega(\gamma^*, \omega^*) \end{bmatrix}.$$

With $\begin{bmatrix} z_1(\gamma^*, \omega^*) & z_2(\gamma^*, \omega^*) \end{bmatrix}^H = Jx^* = y^*$ (see (2.10)) and (2.5), (3.8) we have that

$$f_{\omega\omega}(\gamma^*, \omega^*) = 2{y^*}^H \hat{x}^* \neq 0.$$

Furthermore, using Taylor series expansion near (γ^*, ω^*), we have

$$\gamma(\omega) = \gamma^* - (\omega - \omega^*)^2 \frac{f^*_{\omega\omega}}{2 f^*_\gamma} + \mathcal{O}((\omega - \omega^*)^3).$$

From (3.3) we have that $f^*_\gamma > 0$ and as we are searching for a global maximum we must have $\gamma''(\omega^*) = -\frac{f^*_{\omega\omega}}{f^*_\gamma} \leq 0$. Thus (3.5) follows. □

Both the results (3.3) and (3.5) are evident in the numerical computations in section 5.

Lemma 3.1 shows that the solution structure of $f(\gamma, \omega) = 0$ (and hence of $\det(\mathcal{H}(\gamma, \omega)) = 0$) is as in Figure 1.

Thus for $\gamma < \gamma^*$ there are two (real) values of ω such that $f(\gamma, \omega) = 0$ (see Figure 1). These correspond to algebraically simple eigenvalues of $\mathcal{H}(\gamma)$ on the imaginary

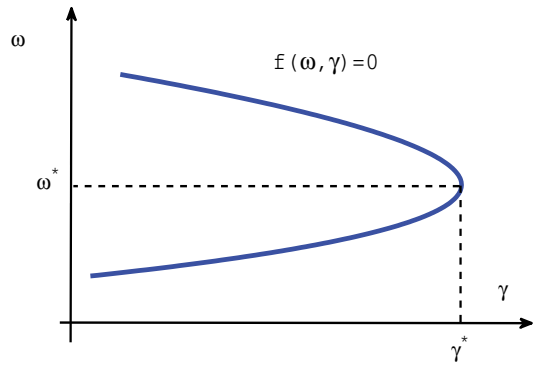

FIG. 1. *Curve $f(\gamma,\omega)=0$ in the (γ,ω)-plane.*

axis. For $\gamma > \gamma^*$ there are no real solutions (and hence $\mathcal{H}(\gamma)$ has no eigenvalues on the imaginary axis, (cf. Theorem 2.1)).

Equation (3.2) and Lemma 3.1 show that we need to solve a nonlinear system

$$g(\gamma,\omega) = \begin{bmatrix} f(\gamma,\omega) \\ f_\omega(\gamma,\omega) \end{bmatrix} = 0 \tag{3.10}$$

in order to find the point (γ^*,ω^*) where $\mathcal{H}(\gamma^*,\omega^*)$ has a two-dimensional Jordan block corresponding to the eigenvalue zero.

In the next section we give details of how to solve this system.

4. Computation of $\|G\|_\infty$. As described in section 2 and 3 above, the optimization problem (1.2), which if treated directly requires the repeated solution of singular values of a transfer function, is reduced to the solution of two nonlinear equations in two real variables given by (3.10). In (3.10), the components of g, namely the real f and f_ω are found by solving the linear systems (3.1) and (3.7), which have the same system matrix given by (2.9)

Hence, in order to find $\|G\|_\infty$ we need to solve a nonlinear problem in two variables and we may use any locally or globally convergent routine to solve the nonlinear system (3.10). In this paper we choose Newton's method, because the required derivatives of f are obtained by a natural and efficient process: With a starting guess $(\gamma^{(0)},\omega^{(0)})$ we need to solve the sequence of linear systems

$$\begin{bmatrix} f_\omega(\gamma^{(i)},\omega^{(i)}) & f_\gamma(\gamma^{(i)},\omega^{(i)}) \\ f_{\omega\omega}(\gamma^{(i)},\omega^{(i)}) & f_{\omega\gamma}(\gamma^{(i)},\omega^{(i)}) \end{bmatrix} \begin{bmatrix} \Delta\gamma^{(i)} \\ \Delta\omega^{(i)} \end{bmatrix} = -g(\gamma^{(i)},\omega^{(i)})$$

for $i = 0,1,2\ldots$ where the function g is given by (3.10). We then update $\left[\begin{smallmatrix}\gamma^{(i+1)}\\ \omega^{(i+1)}\end{smallmatrix}\right] = \left[\begin{smallmatrix}\gamma^{(i)}\\ \omega^{(i)}\end{smallmatrix}\right] + \left[\begin{smallmatrix}\Delta\gamma^{(i)}\\ \Delta\omega^{(i)}\end{smallmatrix}\right]$ for $i = 0,1,2,\ldots$ until convergence. The values of $f(\gamma^{(i)},\omega^{(i)})$, $f_\omega(\gamma^{(i)},\omega^{(i)})$, $f_{\omega\omega}(\gamma^{(i)},\omega^{(i)})$, and $f_\gamma(\gamma^{(i)},\omega^{(i)})$ are calculated from solving the systems in (3.1), (3.7), (3.9), and (3.6). The value of $f_{\omega\gamma}(\gamma^{(i)},\omega^{(i)})$ can be calculated from the following system, which is obtained by differentiating (3.7) with respect to

γ:

$$\begin{bmatrix} & & & v_1 \\ & \mathcal{H}(\gamma,\omega) & & v_2 \\ & & & 0 \\ v_1^H & v_2^H & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{\omega\gamma}(\gamma,\omega) \\ f_{\omega\gamma}(\gamma,\omega) \end{bmatrix} = i \begin{bmatrix} J \begin{bmatrix} (z_1)_\gamma(\gamma,\omega) \\ (z_2)_\gamma(\gamma,\omega) \end{bmatrix} \\ -2i\gamma(z_3)_\omega(\gamma,\omega) \\ 0 \end{bmatrix}.$$

Newton's method is well defined as the Jacobian of $g(\gamma,\omega)$ at the root (γ^*,ω^*) given by

$$J(\gamma^*,\omega^*) = \begin{bmatrix} f_\omega(\gamma^*,\omega^*) & f_\gamma(\gamma^*,\omega^*) \\ f_{\omega\omega}(\gamma^*,\omega^*) & f_{\omega\gamma}(\gamma^*,\omega^*) \end{bmatrix} = \begin{bmatrix} 0 & f_\gamma(\gamma^*,\omega^*) \\ f_{\omega\omega}(\gamma^*,\omega^*) & f_{\omega\gamma}(\gamma^*,\omega^*) \end{bmatrix},$$

is nonsingular, since $f_\gamma(\gamma^*,\omega^*) > 0$ as well as $f_{\omega\omega}(\gamma^*,\omega^*) > 0$, where we have used (3.3), (3.4), and (3.5). Note that the formulation of the Jacobian allows us to detect when Assumption 2.2 does not hold: If $(i\omega^*, x^*)$ is a defective eigenpair of $H(\gamma^*)$ of algebraic multiplicity larger than 2 (i.e., 4, 6, etc.), then $f_{\omega\omega}(\gamma^*,\omega^*) = 0$. Hence a small entry in the Jacobian indicates that there is a higher-dimensional Jordan block in a nearby problem. (This will not occur in the generic situation.)

4.1. Staircase reduction. In order to compute f and its derivatives, linear systems with the same Hermitian system matrix $T(\gamma,\omega)$ given by (2.9) need to be solved. For large values of n this can be done efficiently using the reduction of (A,B,C,D) into staircase form (which can be implemented using the staircase algorithm [29]) as follows.

Initially, we reduce the pair (A,B) to staircase form, that is, $(U^HAU, U^HBV) = (\hat{A},\hat{B})$, where $(\hat{A},\hat{B})$ are in upper block staircase form. $\hat{B} \in \mathbb{C}^{n\times p}$ is upper triangular with right aligned matrix, $\hat{A} \in \mathbb{C}^{n\times n}$ is upper block Hessenberg with upper triangular blocks (which are also right aligned). Here, $U \in \mathbb{C}^{n\times n}$ and $V \in \mathbb{C}^{p\times p}$ are unitary matrices (using, for example, Householder transformations). We obtain the new quadruple $(\hat{A},\hat{B},\hat{C},\hat{D}) = (U^HAU, U^HBV, CU, DV)$. Using this transformation and

$$Q = \begin{bmatrix} UP & 0 & 0 & 0 \\ 0 & 0 & U & 0 \\ 0 & V & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{C}^{2n+p+1\times 2n+p+1}, \quad \text{where} \quad P = \begin{bmatrix} 0 & \cdots & 1 \\ & \iddots & \\ 1 & \cdots & 0 \end{bmatrix} \in \mathbb{C}^{n\times n}$$

is an antidiagonal identity matrix and Q is a unitary matrix, we obtain a unitary similarity transform of $T(\gamma,\omega)$ given by

$$Q^HT(\gamma,\omega)Q = \begin{bmatrix} 0 & PU^HBV & PU^H(A-i\omega I)U & PU^Hv_1 \\ V^HB^HUP & V^HD^HDV-\gamma^2I & V^HD^HCU & 0 \\ U^H(A^H+i\omega I)UP & U^HC^HDV & U^HC^HCU & U^Hv_2 \\ v_1^HUP & 0 & v_2^HU & 0 \end{bmatrix},$$

and hence, with $(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (U^H AU, U^H BV, CU, DV)$ we obtain

$$Q^H T(\gamma,\omega)Q = \begin{bmatrix} 0 & P\hat{B} & P\hat{A} - i\omega P & PU^H v_1 \\ \hat{B}^H P & \hat{D}^H\hat{D} - \gamma^2 I & \hat{D}^H\hat{C} & 0 \\ \hat{A}^H P + i\omega P & \hat{C}^H\hat{D} & U^H C^H CU & U^H v_2 \\ v_1^H UP & 0 & v_2^H U & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & \vartriangle & \times & \times \\ \vartriangle & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} 0 & 0 & -i\omega P & 0 \\ 0 & -\gamma^2 I & 0 & 0 \\ i\omega P & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

This matrix is in lower anti-$(p+1)$-Hessenberg form and solves with this matrix only cost $\mathcal{O}(p(2n+p)^2)$ flops. Note that the transformation of (A, B) into staircase form costs $\mathcal{O}(n^2(n+p))$ operations but need only be done once for entire algorithm. The decomposition $Q^H T(\gamma,\omega)Q$ is only done once and for every new (γ,ω) the algorithm requires $\mathcal{O}(p(2n+p)^2)$ operations. Since we apply Newton's method only a few of those iterations are expected.

Compared to this staircase approach a (symmetric) LU factorization of $T(\gamma,\omega)$ requires $\mathcal{O}(2n+p)^3$ operations, which need to be done at every Newton step. The consequent forward and backward substitutions cost $\mathcal{O}(2n+p)^2$ and hence are done relatively cheaply.

Note that transforming (A, B) into staircase form is only advantageous for medium size dense problems. For larger and sparse problems, sparse LU factorizations should be used, as they are more efficient in terms of storage. We present some numerical experiments in section 5.

4.2. Choice of starting values. Newton's method described at the beginning of this section converges quadratically to a solution of (3.10). However, we can only guarantee local convergence and it is, therefore, important to choose a good starting value for the iteration. It is worth investing some effort in a good starting guess in order to obtain convergence to a global solution of (3.10), since, as pointed out earlier, the actual algorithm can be implemented relatively cheaply. Of course, we can always test whether we obtained a global solution and hence computed $\|G\|_\infty$ (rather than a lower bound on $\|G\|_\infty$) by computing all the pure imaginary eigenvalues of the Hamiltonian matrix $M(\gamma)$ in (2.1) at the converged value of γ, which is expensive.

We suggest two possibilities of creating a starting guess:

(a) We compute a few (say k) rightmost eigenvalues of A and denote their imaginary parts by $s_1, \ldots, s_k$. Then we substitute those values into the transfer function and compute $\gamma^{(0)} = \max\{\sigma_{\max}(D), \max_k \sigma_{\max}\{C(s_k I - A)^{-1}B + D\}\}$, the starting guess for γ. For $\omega^{(0)}$ use the absolute value of the imaginary part of the eigenvalue of $H(\gamma^{(0)})$ closest to the rightmost eigenvalue of A.

(b) We compute a few (say k) dominant poles of the system (1.1) using the subspace accelerated MIMO dominant pole algorithm of Rommes and Martins [26, 25] and denote their imaginary parts by $s_1, \ldots, s_k$. Then we proceed as before and compute $\gamma^{(0)} = \max\{\sigma_{\max}(D), \max_k \sigma_{\max}\{C(s_k I - A)^{-1}B + D\}\}$, the starting guess for γ. For $\omega^{(0)}$ use the absolute value of the imaginary part of the eigenvalue of $H(\gamma^{(0)})$ closest to the rightmost eigenvalue of A.

Both methods give good initial guesses, with the second method particularly suitable for descriptor systems. However, we note that calculating the dominant poles of a system (A, B, C, D) is (generally) more costly than calculating the rightmost eigenvalues of S.

5. Numerical examples. We use examples from [7] and [19] to test the algorithm. For the initial guess we use the choices suggested in section 4.2. For bordering the matrix we need to choose $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right] \in \mathbb{C}^{2n}$ such that (2.8) in Theorem 2.4 is satisfied. A good choice in the limit would be $v = Jx^*$ since $\|x^*\| \neq 0$. We select $v = Jx^{(0)}$, where $x^{(0)}$ is the eigenvector of $H(\gamma^{(0)})$ from (2.2) corresponding to $\omega^{(0)}$, the eigenvalue closest to the rightmost eigenvalue of A, which, in the light of (2.3), is a reasonable choice. First, we consider the convergence for a few simple examples before we compare the staircase algorithm versus a LU decomposition in section 5.1, and, in section 5.2, the algorithm suggested in [19] versus our method. We will also show, in section 5.3, that our method works for descriptor systems, too.

For all our tests we stop the iteration once $\|g\|$ is small enough or once the difference between two consecutive values of γ is small enough, that is,

$$|\gamma^{(i+1)} - \gamma^{(i)}| \leq 10^{-12} \quad \text{or} \quad \|g(\gamma^{(i)}, \omega^{(i)})\| \leq 10^{-12},$$

and note that, at convergence, the value of γ gives the required $\gamma^* = \|G\|_\infty$, if the starting guess is close enough.

EXAMPLE 5.1. *Let*

$$A = \begin{bmatrix} -0.08 & 0.83 & 0 & 0 \\ -0.83 & -0.08 & 0 & 0 \\ 0 & 0 & -0.7 & 9 \\ 0 & 0 & -9 & -0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.4 & 0 & 0.4 & 0 \\ 0.6 & 0 & 1 & 0 \end{bmatrix}, \quad \textit{and} \quad D = \begin{bmatrix} 0.3 & 0 \\ 0 & -0.15 \end{bmatrix}.$$

As starting values we use the suggestion in (a) *of section* 4.2 *with* $k = 1$ *(that is, using the imaginary part of the leftmost eigenvalue).*

TABLE 1
Results for Example 5.1.

i	$\omega^{(i)}$	$\gamma^{(i)}$	$\|g(\gamma^{(i)}, \omega^{(i)})\|$	$f_{\omega\omega}(\omega^{(i)}, \omega^{(i)})$
0	8.3000000000e-01	6.4334694784e+00	-	-
1	8.3374138804e-01	6.4475687221e+00	4.1298665856e+00	1.0941e+03
2	8.3374207344e-01	6.4405007981e+00	1.3502985035e-02	1.1014e+03
3	8.3374207184e-01	6.4405165312e+00	3.0127227283e-05	1.1030e+03
4	8.3374207184e-01	6.4405165313e+00	1.4969643126e-10	1.1030e+03
5	8.3374207184e-01	6.4405165313e+00	1.9577418966e-14	1.1030e+03

The results for Example 5.1 are given in Table 1. We see quadratic convergence of Newton's method to the correct H_∞-norm of this system given by $\gamma^* = \|G\|_\infty = 6.4405$. We also see that $f_{\omega\omega}(\omega^*, \omega^*) > 0$ as predicted by (3.5). We note that we obtain the same results when we use the starting guess from part (b) of section 4.2.

EXAMPLE 5.2. *Consider example* HE7 *from Compleib, a database of continuous-time control-design examples* [24]. *This is a dense matrix example which arises from an application in Helicopter modeling and which was used in* [19] *with* $n = 23$, $m = 16$, *and* $p = 9$. *As starting values we use the suggestion in* (a) *of section* 4.2 *with* $k = 1$.

Table 2 shows the results for Example 5.2. The algorithm converges to $\gamma^* = \|G\|_\infty = 3.4653e + 02$. We observe that, numerically, convergence is only superlinear for this example. We find that the value of $f_\gamma(\omega, \gamma) = \mathcal{O}(10^{-9})$ for this example,

TABLE 2
Results for Example 5.2.

i	$\omega^{(i)}$	$\gamma^{(i)}$	$\|g(\gamma^{(i)},\omega^{(i)})\|$	$f_{\omega\omega}(\omega^{(i)},\omega^{(i)})$
0	1.8344403730e-04	3.4652985979e+02		-
1	-2.4213040169e-05	3.4732871957e+02	2.8008459167e-06	1.2149e-02
2	2.5782034876e-06	3.4650701002e+02	3.6852847834e-07	1.5215e-02
3	7.6832759609e-09	3.4652996280e+02	4.3788829014e-08	1.6999e-02
4	-1.0361942022e-13	3.4652985979e+02	1.3039689593e-10	1.6948e-02
5	1.5991019956e-17	3.4652985979e+02	1.7554869762e-15	1.6948e-02

hence we cannot observe the expected quadratic convergence. From (3.3) we know that $f_\gamma(\omega,\gamma) > 0$; however, if $\|z_3(\omega,\gamma)\|$ is small, then $f_\gamma(\omega,\gamma)$ is small. Note that $\|z_3(\omega,\gamma)\| \neq 0$, unless A has an eigenvalue on the imaginary axis (as can be seen from (3.1)), a case we exclude as we assume that A is stable.

Moreover, we note that for this example computing the dominant poles (option (b)) for the starting values requires the computation of at least three dominant poles (i.e., $k = 3$) in order to obtain convergence to the correct value of $\|G\|_\infty = 3.4653e{+}02$. If we compute only one dominant pole as a starting guess, the algorithm converges to a local minimum and we get only a lower bound for $\|G\|_\infty$.

Next, we consider a significantly larger example.

EXAMPLE 5.3. *Consider example* MM18 *from Compleib. This is a sparse matrix example which was used in* [19] *with* $n = 1006$, $m = 2$, *and* $p = 1$. *As an initial guess we use the suggestion in* (a) *of section* 4.2 *with* $k = 2$.

TABLE 3
Results for Example 5.3.

i	$\omega^{(i)}$	$\gamma^{(i)}$	$\|g(\gamma^{(i)},\omega^{(i)})\|$	$f_{\omega\omega}(\omega^{(i)},\omega^{(i)})$
0	1.0012518357e+02	1.0167782423e+00	-	-
1	1.0001089761e+02	1.0298162308e+00	2.2782486083e-03	1.9984e-02
2	1.0001104318e+02	1.0232992978e+00	1.2494957214e-04	1.9985e-02
3	1.0001104318e+02	1.0233605181e+00	1.1966426790e-06	1.9985e-02
4	1.0001104318e+02	1.0233605236e+00	1.0737835409e-10	1.9985e-02
5	1.0001104318e+02	1.0233605236e+00	1.2727679660e-16	1.9985e-02

TABLE 4
Results for Example 5.3 *using dominant pole as initial guess.*

i	$\omega^{(i)}$	$\gamma^{(i)}$	$\|g(\gamma^{(i)},\omega^{(i)})\|$	$f_{\omega\omega}(\omega^{(i)},\omega^{(i)})$
0	1.0012518357e+02	1.0167782423e+00	-	-
1	1.0001089761e+02	1.0298162308e+00	2.2782486081e-03	1.9984e-02
2	1.0001104318e+02	1.0232992978e+00	1.2494957213e-04	1.9985e-02
3	1.0001104318e+02	1.0233605181e+00	1.1966426787e-06	1.9985e-02
4	1.0001104318e+02	1.0233605236e+00	1.0737835284e-10	1.9985e-02
5	1.0001104318e+02	1.0233605236e+00	1.2785069419e-16	1.9985e-02

The results for Example 5.3, which show quadratic convergence to $\gamma^* = \|G\|_\infty = 1.0234$ are given in Table 3. For this example, using option (b) of section 4.2 for the starting values (the dominant poles) with $k = 1$ (that is, calculating only one dominant pole) gives the results in Table 4; that is, similar fast quadratic convergence but only one dominant pole is necessary to create the initial guess.

We note that for all examples convergence of the algorithm is very fast and we

obtain the expected H_∞-norm of the system. However, for these examples we need a starting guess close to the solution as Newton's method might converge to a lower bound on $\|G\|_\infty$ if a bad starting value is chosen.

5.1. Staircase algorithm. In this subsection we compare dense examples using the staircase algorithm suggested in section 4.1. This approach should be faster than an LU decomposition, especially when medium size dense problems are used. To this end we create random (stable) matrices of size $n = 80, 160, 240, \ldots, 1200$ and bring them into upper Hessenberg form. For simplicity we chose $m = p = 1$ and $D = 0$ in (1.1). The performance of the algorithm using the staircase reduction (and hence a solve with an anti-$p+1$-Hessenberg form) and the standard LU decomposition (performed at every step) are then compared.

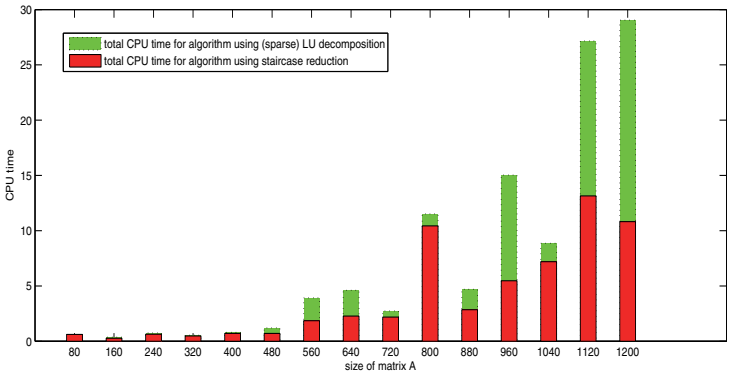

FIG. 2. *Comparison of CPU time for algorithm using the staircase form described in section* 4.1 *and algorithm using a (sparse) LU decomposition.*

The results showing the total CPU times using either (sparse) LU decompositions or the staircase form are plotted in Figure 2. We observe that the implicit determinant method using the staircase algorithm performs better in terms of CPU time than using a LU factorization at every step of the Newton method. The improvement of performance is particularly seen for larger problems.

5.2. Comparison with spectral value sets method. Finally, we compare the algorithm described in this paper (the implicit determinant method, or IDM) with the method introduced in [19], which uses spectral value sets to find the H_∞-norm. That code was available from [22]. We label that method SVS for spectral value sets. We compare both methods for a range of examples from Compleib [24] and EigTool [31] which were also considered in [19]. The results for a range of small dense problems as well as larger sparse problems are shown in Table 5.

The table shows results for computing the H_∞-norm using (i) MATLAB, (ii) the implicit determinant method (IDM), and (iii) spectral value sets (SVS). The first four columns describe the problems and their sizes. The fifth column gives the result for the H_∞-norm using the implicit determinant method, the sixth column the number k of rightmost eigenvalues computed to obtain a starting value, and the seventh column the number of Newton iterations for the IDM. The eighth and ninth column show the result for the H_∞-norm using MATLAB and spectral value sets, the tenth and eleventh column give the corresponding CPU times. The CPU times are showing the

TABLE 5

Comparison of performance of the IDM and the method using spectral value sets (SVS) introduced in [19].

	n	m	p	k	$\|\|G\|\|_\infty^{IDM}$	NI	$\|\|G\|\|_\infty^{\text{MATLAB}}$	$\|\|G\|\|_\infty^{SVS}$	CPU(IDM)	CPU(SVS)
BBK	4	2	2	1	$6.44051653e+00$	5	$6.41141714e+00$	$6.44051653e+00$	3.000e-01	**2.100e-01**
CBM	351	2	1	5	$2.62976526e-01$	5	$2.62963461e-01$	$2.62976526e-01$	**3.100e+00**	1.106e+02
CSE2	63	32	1	3	$2.03391753e-02$	4	$2.03373479e-02$	$2.03391753e-02$	**5.400e-01**	8.238e+01
CM1	23	3	1	7	$8.16496496e-01$	5	$8.16496495e-01$	$8.16496589e-01$	**2.100e-01**	5.700e-01
CM3	123	3	1	5	$8.16094047e-01$	7	$8.18627162e-01$	$8.21443671e-01$	**3.900e-01**	1.687e+01
CM4	243	3	1	6	$1.58866664e+00$	5	$1.58120303e+00$	$1.44542364e+00$	**1.360e+00**	6.475e+02
HE6	23	16	6	3	$4.92937305e+02$	1	$4.92937305e+02$	$4.92937159e+02$	**4.800e-01**	8.430e+00
HE7	23	16	9	1	$3.46529860e+02$	5	$3.46529860e+02$	$3.46529860e+02$	**3.100e-01**	7.800e-01
ROC1	12	2	2	3	$1.21658902e+00$	5	$1.21710734e+00$	$1.21658902e+00$	**2.100e-01**	4.200e-01
ROC2	13	1	4	1	$1.33366743e-01$	2	$1.33366743e-01$	$1.33366743e-01$	**3.600e-01**	3.900e-01
ROC3	14	11	11	1	$1.72310471e+04$	4	$1.72308749e+04$	$1.72310471e+04$	**3.000e-01**	3.600e-01
ROC4	12	2	2	1	$2.95650873e+02$	8	$2.94963731e+02$	$2.95650873e+02$	**1.900e-01**	3.500e-01
ROC5	10	2	3	1	$9.79995384e-03$	1	$9.79995303e-03$	$9.79995186e-03$	**2.800e-01**	6.200e-01
ROC6	8	3	3	1	$2.57633040e+01$	1	$2.57633040e+01$	$2.57633040e+01$	3.400e-01	**2.200e-01**
ROC7	8	3	1	2	$1.12196212e+00$	9	$1.12196205e+00$	$1.12196215e+00$	**1.000e-01**	1.420e+00
ROC8	12	7	1	1	$6.59896425e+00$	5	$6.59893055e+00$	$6.59896425e+00$	2.800e-01	**2.700e-01**
ROC9	9	5	1	3	$3.29185918e+00$	6	$3.27373249e+00$	$3.29404544e+00$	**1.300e-01**	5.000e-01
ROC10	9	2	2	1	$1.01446326e-01$	4	$1.01467934e-01$	$1.01480433e-01$	**3.300e-01**	4.500e-01
NN18	1006	2	1	2	$1.02336052e+00$	5	-	$1.02336052e+00$	**3.607e+01**	4.604e+01
dwave	2048	6	4	1	$3.80199635e+04$	1	-	$3.80199635e+04$	**1.516e+01**	1.077e+02
pde	2961	6	4	2	$3.68749948e+02$	11	-	$3.68749948e+02$	2.124e+02	**6.510e+01**

full timings, including the time for computing the initial guess (for the IDM we use the option (a) suggested in section 4.2). The faster times are written in bold in the last two columns of Table 5.

First, we note that for problems `CM3` and `CM4` both our methods struggle to compute the correct value for the H_∞-norm; however, the error with the IDM appears to be slightly smaller. We also observe that, for small to medium size problems (the problems above the line in Table 5), in for 15 out of 18 examples, the IDM is faster than the method using SVS. For a few small problems the method using SVS is faster, but the CPU times are comparable. Note that, in particular, for problems with larger values of n (but still medium size problems), that is, `CBM`, `CM3`, and `CM4`, the gain in terms of CPU time is significant.

For larger, sparse problems (the problems below the line in Table 5) the IDM still outperforms the method using SVS, up to a certain size of the system. For very large problems, it is not practical to use the LU factorization or staircase form any more as storage and computation time prevents the method from being efficient. For very large and sparse problems the spectral value set method from [19] outperforms our algorithm. For these problems the linear systems that arise in the IDM would need to be solved iteratively, using, for example, efficient preconditioned Krylov subspace methods, in order to obtain satisfactory performance of the IDM. This will be the subject of future research.

5.3. Extension to descriptor systems. Finally, we present an example for descriptor systems, that is, systems of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

where $E \neq I$. One such system is `peec`, a model arising from a partial element equivalent circuit model of a patch antenna structure [11]. The system has $n = 480$, $m = 1$, and $p = 1$. For this particular problem E is singular. As an initial guess, we use the suggestion in (b) of section 4.2, that is, the dominant pole algorithm. If we compute $k = 10$ dominant poles we obtain $\omega^{(0)} = 5.4635$ and $\gamma^{(0)} = 3.5252$ and the results for the implicit determinant method are shown in Table 6.

TABLE 6
Results for descriptor system `peec` *from the SLICOT benchmark collection.*

i	$\omega^{(i)}$	$\gamma^{(i)}$	$\|g(\gamma^{(i)},\omega^{(i)})\|$	$f_{\omega\omega}(\omega^{(i)},\omega^{(i)})$
0	5.4634925189e+00	3.5252017504e-01	-	-
1	5.4634927117e+00	3.5265371901e-01	5.5856558957e-06	2.8947e+01
2	5.4634927118e+00	3.5259605815e-01	1.6866054688e-11	2.8947e+01
3	5.4634927118e+00	3.5259607234e-01	4.9581046141e-13	2.8947e+01

We observe quadratic convergence to $\gamma^* = \|G\|_\infty = 3.5260e-01$. Note that a good initial guess is very important for this example. If we choose $k < 10$ dominant poles to compute a starting guess, then the algorithm converges to the wrong local maximum. Moreover, if we had chosen the method (a) in of section 4.2 to compute the starting guess (that is, compute k rightmost eigenvalues of the generalized eigenproblem (A, E)), we would have had to calculate $k \geq 97$ rightmost eigenvalues in order to get a good enough starting value for convergence to the correct value of γ^*. The `peec` problem is a hard problem in the sense that the transfer function has a lot of peaks due to many poles close to the imaginary axis. The maximum peak (which we are looking for) is very spiky and thin. Therefore, it requires the computation of many poles ($k = 10$) and even more rightmost eigenvalues($k \geq 97$) to find a starting value which converges to the global solution. For details of this specific problem, we refer to [5].

6. Conclusions. We have introduced a new iterative scheme to compute the H_∞-norm of a transfer function. The method relies on (i) the relationship between singular values of a transfer function matrix and pure imaginary eigenvalues of a Hamiltonian matrix and (ii) the efficient computation of two-dimensional Jordan blocks in a parameter-dependent system.

We have given several numerical examples that show the performance of the method. We have also seen that the method can be applied to descriptor systems. We have shown that for small and medium size problems the method outperforms algorithms recently introduced in [19, 5]. However, for large, sparse problems, the methods in [19, 5] for the approximation of the H_∞-norm are faster, as they use iterative eigensolvers. We reduce the eigensystems to be solved to linear systems but solve them using either (sparse) factorizations or a reduction of the system to staircase form. In order to compete with the methods in [19, 5] for larger problems we will have to use efficient preconditioned Krylov subspace methods, in order to obtain satisfactory performance of the implicit determinant method.

We remark that generally our method provides local convergence only, hence only a lower bound on the H_∞-norm can be computed. Therefore, the strategy of computing good starting values for the algorithm is important. We explored the heuristic of using a number of rightmost points and dominant poles, which both provide good starting values. (We observe the latter ones are often better, in particular for descriptor systems.) We also observe that the more of these values we compute initially to find a good starting guess, the better the initial value is. We are not able to guarantee convergence to the global minimum, but in almost all our computations our algorithm obtained the correct value.

We note that the algorithm introduced here can also be used for discrete time linear dynamical systems, as was shown in [21] for the discrete distance to instability.

Acknowledgments. The first author gratefully acknowledges the hospitality of Raf Vandebril (KU Leuven) and Paul Van Dooren (UCL) for a funded visit to KU Leuven during which this work was finalized. The first author also thanks Matthias Voigt (MPI Magdeburg) and Michael Overton (NYU) for helpful discussions on this work.

REFERENCES

[1] R. O. Akinola, M. A. Freitag, and A. Spence, *The calculation of the distance to a nearby defective matrix*, Numer. Linear Algebra Appl., 21 (2014), pp. 403–414.

[2] R. O. Akinola, M. A. Freitag, and A. Spence, *The computation of Jordan blocks in parameter-dependent matrices*, IMA J. Numer. Anal., in press.

[3] R. Alam, S. Bora, M. Karow, V. Mehrmann, and J. Moro, *Perturbation theory for Hamiltonian matrices and the distance to bounded-realness*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 484–514.

[4] P. Benner, R. Byers, V. Mehrmann, and H. Xu, *Numerical methods for linear quadratic and h_∞ control problems*, in Dynamical Systems, Control, Coding, Computer Vision, Birkhäuser, Basel, 1999, pp. 203–222.

[5] P. Benner and M. Voigt, *A Structured Pseudospectral Method for h_∞-norm Computation of Large-scale Descriptor Systems*, Max Planck Institute Magdeburg, preprint, 2012.

[6] 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 (1990), pp. 1–7.

[7] S. Boyd, V. Balakrishnan, and P. Kabamba, *A bisection method for computing the H_∞-norm of a transfer matrix and related problems*, Math. Control Signals Systems, 2 (1989), pp. 207–219.

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

[9] 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.

[10] 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.

[11] Y. Chahlaoui and P. Van Dooren, *Benchmark examples for model reduction of linear time invariant dynamical systems*, P. Benner et al, eds., Lect. Notes Comput. Sci. Eng. 45, Springer, Berlin, Heidelberg, New York, 2002.

[12] B. M. Chen, *Robust and H_∞ Control*, Springer, New York, 2000.

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

[14] M. A. Freitag and A. Spence, *A new approach for calculating the real stability radius*, BIT Numer. Math., in press.

[15] Y. Genin, P. V. Dooren, and V. Vermaut, *Convergence of the calculation of H_∞-norms and related questions*, in Proceedings of the Conference on the Mathematical Theory of Networks and Systems, MTNS '98, A. Beghi, L. Finesso, and G. Picci, eds., 1998, pp. 429–432.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, 1996.

[17] A. Griewank and G. W. Reddien, *Characterization and computation of generalized turning points*, SIAM J. Numer. Anal., 21 (1984), pp. 176–185.

[18] S. Gugercin and A. C. Antoulas, *A survey of model reduction by balanced truncation and some new results*, Internat. J. Control, 77 (2004), pp. 748–766.

[19] 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 (2013), pp. 709–737.

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

[21] M. Gürbüzbalaban, *Theory and Methods for Problems Arising in Robust Stability, Optimization and Quantization*, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, New York, 2012.

[22] M. Gürbüzbalaban, *Hinfnorm*. see http://cims.nyu.edu/∼mert/software/hinfinity.html, 2013.

[23] H. B. Keller, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359–384.
[24] F. Leibfritz, *COMPLEIB: Constrained Matrix Optimization Problem Library*, 2006.
[25] J. Rommes, *Methods for Eigenvalue Problems with Applications in Model Order Reduction*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 2007.
[26] J. Rommes and N. Martins, *Efficient computation of multivariable transfer function dominant poles using subspace acceleration*, IEEE Trans. Power Syst., 21 (2006), pp. 1471–1483.
[27] A. Spence and I. G. Graham, *Numerical methods for bifurcation problems*, in The Graduate Student's Guide to Numerical Analysis '98 (Leicester), Springer Ser. Comput. Math. 26, Springer, Berlin, 1999, pp. 177–216.
[28] A. Spence and C. Poulton, *Photonic band structure calculations using nonlinear eigenvalue techniques*, J. Comput. Phys., 204 (2005), pp. 65–81.
[29] P. M. Van Dooren and M. Verhaegen, *On the use of unitary state-space transformations*, Contemp. Math., 47 (1985), pp. 447–463.
[30] C. F. Van Loan, *How near is a stable matrix to an unstable matrix?*, in Linear Algebra and Its Role in Systems Theory (Brunswick, ME, 1984), Contemp. Math., 47 (1985), pp. 465–478.
[31] T. G. Wright, *Eigtool*, see http://www.comlab.ox.ac.uk/pseudospectra/eigtool/, 2002.