Fast algorithms for computing the distance to instability of nonlinear eigenvalue problems, with application to time-delay systems

Dries Verhees · **Roel Van Beeumen** · **Karl Meerbergen** · **Nicola Guglielmi** · **Wim Michiels**

Abstract A continuous dynamical system is stable if all eigenvalues lie strictly in the left half of the complex plane. However, this is not a robust measure because stability is no longer guaranteed when the system parameters are slightly perturbed. Therefore, the pseudospectrum of a matrix and its pseudospectral abscissa are studied. Mostly, one is often interested in computing the distance to instability, because it is a robust measure for stability against perturbations. As a first contribution, this paper presents two iterative methods for computing the distance to instability, considering complex perturbations. The first one is based on locating a zero of the pseudospectral abscissa function. This method is particularly suitable for large sparse matrices as it is based on repeated eigenvalue computations, where the original matrix is perturbed with a rank one matrix. The second method is based on a recently proposed global optimization technique. The advantages of both methods can be combined in a hybrid algorithm. As a second contribution we show that the methods apply to a broad class of nonlinear eigenvalue problems, in particular eigenvalue problems inferred from linear delay-differential equations, and, therefore, they are useful for a wide range of problems. In the numerical examples the standard eigenvalue problem, the quadratic eigenvalue problem and the delay eigenvalue problem are addressed.

D. Verhees
E-mail: `driesverhees@gmail.com`

N. Guglielmi
Dipartimento di Ingegneria Scienze Informatiche e Matematica, Universita' degli Studi di L' Aquila Via Vetoio - Loc. Coppito, 67010 L' Aquila, Italy.
E-mail: `guglielm@dm.univaq.it`

R. Van Beeumen, K. Meerbergen, W. Michiels
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium.
E-mail: `{Roel.VanBeeumen,Karl.Meerbergen,Wim.Michiels}@cs.kuleuven.be`

1 Introduction

A given linear dynamical system $\dot{x} = Ax$, with $A \in \mathbb{R}^{n \times n}$ is exponentially stable, if and only if all eigenvalues have a strictly negative real part. Or equivalently, the *spectral abscissa* of A, which is the maximum of the real parts of the eigenvalues, is negative. Unfortunately, the spectral abscissa is not a robust measure for stability. This paper focuses on the effects of perturbations on the stability of a given dynamical system. Therefore the *ε-pseudospectrum* $\Lambda_{\varepsilon}(A)$ is studied, i.e. the set of the eigenvalues of all real or complex matrices in a given neighborhood of A. In particular,

$$\Lambda_{\varepsilon}^{\mathbb{K}, \|\cdot\|}(A) := \left\{ \lambda \; : \; \lambda \in \Lambda(B), \|A - B\| \leq \varepsilon, B \in \mathbb{K}^{n \times n} \right\}, \tag{1}$$

with $\Lambda(B)$ the spectrum of B, $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$ and $\|\cdot\|$ a chosen norm [23]. In this paper, we address complex perturbations measured with the Euclidean norm. Therefore, in order not to overload the notation, we will denote $\Lambda_{\varepsilon}^{\mathbb{C}, \|\cdot\|_2}(A)$ simply by $\Lambda_{\varepsilon}(A)$. The largest possible real part of points in the pseudospectrum is called the *ε-pseudospectral abscissa* $\alpha_{\varepsilon}(A)$, i.e.,

$$\alpha_{\varepsilon}(A) := \max\left\{ \mathrm{Re}(\lambda) \; : \; \lambda \in \Lambda_{\varepsilon}(A) \right\}. \tag{2}$$

Perturbations with bound ε may render the system unstable whenever $\alpha_{\varepsilon}(A) \geq 0$. To guarantee stability, the norm of perturbations ε has to be limited. The distance to instability $\mu(A)$ is a strict upper bound on the norm of perturbations such that all perturbed systems are stable. In particular, it holds that $\alpha_{\varepsilon}(A) < 0$ if and only if $\varepsilon < \mu(A)$. One could easily see that $\varepsilon = \mu(A)$ is the critical value for which $\alpha_{\varepsilon}(A) = 0$. Most earlier proposed methods for computing this distance require in each iteration the computation of all eigenvalues on the imaginary axis of a $2n \times 2n$ Hamiltonian matrix [2,3,15]. Even though convergence to the correct solution is guaranteed, these methods are not computationally tractable when n is large. Recently in [6], a fast local method has been described based on computing a Jordan block of the corresponding Hamiltonian matrix by Newton's method. In this paper, we propose a fast local method grounded in the pseudospectral abscissa computation, more precisely in the iteration proposed in [9,17], as well as a global method based on the optimization algorithm of [14]. In addition, we show that these algorithms apply in fact to a broad class of nonlinear eigenvalue problems, described by

$$\left(\sum_{i=0}^{m} A_i p_i(\lambda) \right) v = 0, \tag{3}$$

where $A_0, \ldots, A_m$ are real n-by-n matrices and $p_i : \mathbb{C} \to \mathbb{C}$ are entire functions satisfying $p_i(\bar{\lambda}) = \overline{p_i(\lambda)}$, $i = 1, \ldots, m$. The latter assumption guarantees that eigenvalues are either real or appear as complex conjugate pairs. The class (3) includes polynomial eigenvalue problems and nonlinear eigenvalue problems arising in the analysis of linear delay differential equations as special cases. Although many of these problems can be reformulated as linear eigenvalue problems (for example, by a so-called linearization of a polynomial eigenvalue problem or by a transformation to an equivalent operator eigenvalue problem for the delay eigenvalue problem [19]), we will

not consider the unstructured pseudospectra of a particular type of linearization (note that the pseudospectra strongly depend on the basis). Instead, as in [16,22], we will explicitly take the structure of the original nonlinear eigenvalue problem into account in the definition of pseudospectra, which we relate to the effect of perturbations on the individual coefficient matrices.

2 Preliminaries

2.1 Pseudospectra and distance to instability for nonlinear eigenvalue problems

We consider the nonlinear eigenvalue problem (3). In what follows we call

$$F(\lambda) := \sum_{i=0}^{m} A_i p_i(\lambda) \tag{4}$$

the characteristic matrix. We call the spectrum Λ and the spectral abscissa α, i.e.,

$$\Lambda(F) := \left\{ \lambda \in \mathbb{C} : \det\left(\sum_{i=0}^{m} A_i p_i(\lambda) \right) = 0 \right\}, \tag{5}$$

$$\alpha(F) := \sup\left\{ \mathrm{Re}\,(\lambda) : \lambda \in \Lambda(F) \right\}. \tag{6}$$

We are interested in the effect of bounded perturbations of the matrices A_i on the spectrum, which leads to the perturbed eigenvalue problem

$$\left(\sum_{i=0}^{m} (A_i + \delta A_i) p_i(\lambda) \right) v = 0, \qquad \lambda \in \mathbb{C},\ v \in \mathbb{C}^n \setminus \{0\}. \tag{7}$$

The first step in the robustness analysis is to define the class of perturbations under consideration, as well as a measure of the combined perturbation

$$\Delta := (\delta A_0, \ldots, \delta A_m).$$

In analogy to the classical definition of ε-pseudospectrum of a matrix [23], we allow the perturbations δA_i, $i = 0, \ldots, m$, to be *complex* matrices, i.e.,

$$\Delta \in \mathbb{C}^{n \times n \times (m+1)}. \tag{8}$$

Introducing weights $w_i \in \mathbb{R}_0^+ \cup \{\infty\}, i = 0, \ldots, m$, we define the following global measure of the perturbations:

$$\|\Delta\|_{\mathrm{glob}} := \left\| \begin{bmatrix} w_0 \|\delta A_0\|_2 \\ \vdots \\ w_m \|\delta A_m\|_2 \end{bmatrix} \right\|_{\infty}. \tag{9}$$

In this way, the condition

$$\|\Delta\|_{\mathrm{glob}} \leq \varepsilon$$

corresponds to the natural assumptions of taking perturbations satisfying

$$\|\delta A_i\|_2 \leq \varepsilon/w_i, \ i = 0, \ldots, m.$$

This uncertainty bound is also used in [22] and fits within the general class considered in [16]. Setting a weight equal to infinity implies that the corresponding matrix is not perturbed.

With the above class of allowable perturbations and with the measure (9) we define the ε-pseudospectrum of (3) as

$$\Lambda_\varepsilon = \bigcup_{\|\Delta\|_{\mathrm{glob}} \leq \varepsilon} \left\{ \lambda \in \mathbb{C} \ : \ \det\left(\sum_{i=0}^{m} (A_i + \delta A_i)\, p_i(\lambda) \right) = 0 \right\}, \tag{10}$$

The set Λ_ε is closed and its boundary is generically smooth. Moreover if $\varepsilon_2 > \varepsilon_1$, then we have $\Lambda_{\varepsilon_1} \subseteq \Lambda_{\varepsilon_2}$.

The corresponding pseudospectral abscissa is defined as

$$\alpha_\varepsilon(F) := \sup\{\mathrm{Re}(\lambda) : \ \lambda \in \Lambda_\varepsilon\}. \tag{11}$$

In [16] the following explicit expression for the pseudospectra is obtained.

Proposition 1 *For the perturbation class (8) and measure (9) the pseudospectrum Λ_ε of (4) satisfies*

$$\Lambda_\varepsilon(F) = \left\{ \lambda \in \mathbb{C} : \ \sigma_n\left(\sum_{i=0}^{m} A_i p_i(\lambda) \right) \leq \varepsilon \|w(\lambda)\|_1 \right\}, \tag{12}$$

where $\sigma_n(\cdot)$ denotes the smallest singular value and

$$w(\lambda) = \left[\frac{p_0(\lambda)}{w_0} \ \cdots \ \frac{p_m(\lambda)}{w_m} \right]^T. \tag{13}$$

We say that F is exponentially stable if all zeros are confined to the open left half plane and bounded away from the imaginary axis, i.e., $\alpha(F) < 0$. To assess the robustness of stability w.r.t. perturbations on the coefficient matrices A_i we introduce the concept of distance to instability.

Definition 1 The distance to instability of (4) (also called stability radius) w.r.t. the perturbation class (8) and measure (9) is defined as

$$\mu(F) := \inf\{\varepsilon \geq 0 : \ \alpha_\varepsilon(F) \geq 0\}.$$

A comment is necessary here. The perturbation measure considered in the definition of $\Lambda_\varepsilon(F)$ and $\mu(F)$ permits to robustify the notion of stability of the system, thereby exploiting the structure of the characteristic matrix F. It should be noted that, in this way, the pseudospectra and distance to instability depend on the choice of basis functions p_i, $i = 0, \ldots, m$. However, in most applications, a description in a specific basis gives rise to a physical interpretation of the coefficient matrices. For example,

finite-element codes in acoustics and vibrations often lead to quadratic eigenvalue problems expressed in a monomial basis,

$$F(\lambda) = A_0 + A_1\lambda + A_2\lambda^2, \tag{14}$$

where the three coefficient matrices contain information about stiffness, damping and momentum of the system. This information would be merged in a description in another polynomial basis. Similarly, in control problems that give rise to a delay eigenvalue problem, such as

$$F(\lambda) = -\lambda I + A_1 + A_2 e^{-\lambda\tau}, \tag{15}$$

the coefficient matrix corresponding to the exponential term often has an interpretation in terms of inputs, outputs and a feedback control, which is prone to a dead-time in the control loop.

To conclude the section we state and discuss the main assumption made throughout the paper.

Assumption 1 *There exist a number $\varepsilon_{\max} > \mu(F)$ such that for arbitrary but fixed $r \in \mathbb{R}$ and $\varepsilon \in [0, \varepsilon_{\max})$ the set $\Lambda_\varepsilon \cap \{\lambda \in \mathbb{C} : Re(\lambda) \geq r\}$ is bounded.*

For delay eigenvalue problem (15), Assumption 1 is satisfied for any value of $\varepsilon_{\max}$ larger than $\mu(F)$, as long as only A_1 and A_2 are perturbed. For quadratic eigenvalue problem (14) with nonsingular A_2, the assumption is satisfied by taking $\varepsilon_{\max} = w_2\sigma_n(A_2)$, with w_2 the weight on the perturbation of A_2. This bound stems from the fact that for $\varepsilon > w_2\sigma_n(A_2)$ the ε-pseudospectrum is unbounded in all directions in the complex plane, containing perturbations of an eigenvalue at infinity [17, Section 2.2].

Assumption 1 implies that, by varying ε in a continuous way, a transition from the situation where $\alpha_\varepsilon < 0$ to a situation where $\alpha_\varepsilon \geq 0$ is characterized by eigenvalues moving from the open left half plane to the imaginary axis (i.e., right half plane eigenvalues coming from the point at infinity cannot occur). Combined with the characterization (12) this leads to the following expression for the stability radius [16, Corollary 3].

Proposition 2 *If the zeros of F are confined to the open left half plane and Assumption 1 holds then we have*

$$\mu(F) = \left\{ \sup_{\omega \in \mathbb{R}} \left\| \left(\sum_{i=0}^{m} A_i p_i(j\omega) \right)^{-1} \right\|_2 \|w(j\omega)\|_1 \right\}^{-1}, \tag{16}$$

with w given by (13).

3 Previous work

The approach proposed in [17] to compute the ε-pseudospectral abscissa for a nonlinear eigenvalue problem is inspired by [9] where an iterative algorithm is proposed to find a locally rightmost point of the pseudospectrum of a matrix. This algorithm is mainly based on the property that a boundary point of the pseudospectrum is an eigenvalue of $A + \varepsilon u v^*$, with u and v respectively left and right corresponding eigenvectors. This gives rise to a fixed-point iteration method where the left and right eigenvectors determine the next perturbation. In [7] a continuous dynamical version of the algorithm is presented. More precisely, a differential equation on the manifold of normalized rank one matrices is constructed, having as stationary point an optimal perturbation for which the corresponding rightmost eigenvalue is an extremal point of the pseudospectrum. Extensions to real perturbations are described in [8].

3.1 Computing pseudospectral abscissa for large-scale linear and nonlinear eigenvalue problems

In what follows, we review the method of [17], which generalizes the method of [9] to nonlinear eigenvalue problems.

Assumption 1 guarantees the presence of a globally rightmost point of the pseudospectrum Λ_ε, which we denote by λ_ε, i.e.,

$$\mathrm{Re}\,(\lambda_\varepsilon) = \alpha_\varepsilon.$$

Let $(u_\varepsilon, v_\varepsilon)$ be a pair of normalized left and right singular vectors corresponding to

$$\sigma_n \left(\sum_{i=0}^{m} A_i p_i(\lambda_\varepsilon) \right)$$

and let $c \in \mathbb{C}^n$ be such that u_ε satisfies the normalization constraint

$$c^* u_\varepsilon \in \mathbb{R}_0^+.$$

We make some technical assumptions, on which the algorithm relies.

Assumption 2 *The following conditions are assumed:*

1. *the smallest singular value of $\sum_{i=0}^{m} A_i p_i(\lambda_\varepsilon)$ is simple;*
2. *the rightmost eigenvalue of*

$$\left(\sum_{i=0}^{m} \left(A_i - \frac{\overline{p_i(\lambda_\varepsilon)}}{w_i |p_i(\lambda_\varepsilon)|} u_\varepsilon v_\varepsilon^* \varepsilon \right) p_i(\lambda) \right) v = 0 \tag{17}$$

   is simple;
3. *we have $p_i(\lambda_\varepsilon) \neq 0$ whenever w_i is finite, for $0 \leq i \leq m$.*

The cases where Assumption 2 is not satisfied correspond to degenerate cases, see [17] for a detailed discussion. In Propositions 3.1 and 3.3 of this reference it has been shown that under Assumption 2 the triple $(\lambda, u, v) = (\lambda_\varepsilon, u_\varepsilon, v_\varepsilon)$ is an isolated solution of the following system of equations and inequalities:

$$\left(\sum_{i=0}^{m} \left(A_i - \frac{\overline{p_i(\lambda)}}{w_i|p_i(\lambda)|} uv^*\varepsilon \right) p_i(\lambda) \right) v = 0, \tag{18}$$

$$u^* \left(\sum_{i=0}^{m} \left(A_i - \frac{\overline{p_i(\lambda)}}{w_i|p_i(\lambda)|} uv^*\varepsilon \right) p_i(\lambda) \right) = 0, \tag{19}$$

$$u^*u = v^*v = 1, \ c^*u > 0, \tag{20}$$

$$u^* \left(\sum_{i=0}^{m} A_i p_i'(\lambda) \right) v - \varepsilon \sum_{i=0}^{m} \frac{\overline{p_i(\lambda)} p_i'(\lambda)}{w_i|p_i(\lambda)|} > 0. \tag{21}$$

Furthermore, λ_ε is the rightmost eigenvalue of the perturbed eigenvalue problem (17). The meaning of (18)-(21) can be summarized as follows. Conditions (18)-(19) characterize points lying on curves described by

$$\sigma_k \left(\sum_{i=0}^{m} A_i p_i(\lambda) \right) = \varepsilon \|w(\lambda)\|_1, \qquad k \in \{1, \ldots, n\},$$

which include the boundaries of the pseudospectrum (case where $k = n$). Condition (21) is an optimality condition characterizing that the outwards pointing normal vector to the set

$$\left\{ \lambda \in \mathbb{C} : \ \sigma_k \left(\sum_{i=0}^{m} A_i p_i(\lambda) \right) - \varepsilon \|w(\lambda)\|_1 \leq 0 \right\}$$

lies in the direction of the positive real axis. Conditions (20) are normalization constraints.

The algorithms in [17] are based on turning (18)-(21) into a fixed-point iteration. The basic steps are described by Algorithm 1. Note that there may not exist u_k, v_k satisfying (23) and (24), or they may not be unique. However, there exist remedies for this, which are discussed in detail in [17]. Summarizing, if there is no solution, the conditions are relaxed. If there are multiple solutions, the solution is selected that gives rise to the largest value of $\mathrm{Re}(\lambda_{k+1})$.

Refinements of the basic algorithm to enforce global convergence to the globally rightmost point of the pseudospectrum can be found in [17], consisting of an adaptation to make the sequence $\{\mathrm{Re}(\lambda_k)\}_{k \geq 1}$ monotonic (and in this way preventing, for example, that the iterations converge to a non-equilibrium solution), and the possibility to start with several rightmost eigenvalues (to avoid converging to a locally but not globally rightmost point). All variants share the property that their application only requires methods to compute the *rightmost* eigenvalues and the corresponding left and right eigenvectors, for which fast iterative solvers can be used if the system matrices are large and sparse. This feature makes them also applicable to large-scale

---

Algorithm 1 Fixed-point iteration of [17]

Initialize (λ_0, u_0, v_0) and repeat for $k = 1, 2, \dots$

1. Compute λ_k as the rightmost eigenvalue of

$$\left(\sum_{i=0}^{m} (A_i + \delta A_i) p_i(\lambda) \right) v = 0, \tag{22}$$

where

$$\delta A_i = -\frac{\overline{p_i(\lambda_{k-1})}}{w_i |p_i(\lambda_{k-1})|} u_{k-1} v_{k-1}^* \varepsilon, \qquad 0 \le i \le m.$$

2. Define u_k and v_k as the left and right eigenvectors of (22) associated with λ_k, scaled as

$$u_k^* u_k = v_k^* v_k = 1, \qquad c^* u_k > 0, \tag{23}$$

$$u_k^* \left(\sum_{i=0}^{m} A_i p_i'(\lambda_k) \right) v_k - \varepsilon \sum_{i=0}^{m} \frac{\overline{p_i(\lambda_k)} p_i'(\lambda_k)}{w_i |p_i(\lambda_k)|} > 0. \tag{24}$$

---

problems. The local convergence is linear, with a convergence factor tending to zero as $\varepsilon \to 0$. For an extended discussion on this point see [17].

An application of the method to compute the distance to instability has been considered in [18] and is described in Section 4.1.

4 Distance to instability: local method based on root finding of spectral abscissa function

4.1 Newton's method

We start with a technical lemma about the *pseudospectral abscissa function*

$$\mathbb{R}_+ \ni \varepsilon \mapsto \alpha_\varepsilon(F), \tag{25}$$

which is a consequence of Assumption 1 and the characterization (12).

Lemma 1 *The function (25) is continuous, increasing, and, if not all weights are infinity, then we have* $\lim_{\varepsilon \to \infty} \alpha_\varepsilon = \infty$.

Under mild conditions we have in fact that α_ε is strictly increasing (this will be clarified shortly in Theorem 3). Based on Definition 1 and Lemma 1 we can refine (for exponentially stable F)

$$\mu(F) = \inf \{ \varepsilon > 0 : \ \alpha_\varepsilon(F) = 0 \},$$

and in case of strict monotonicity of α_ε we can drop the infimum, i.e., we can compute $\mu(F)$ as the value ε such that $\alpha_\varepsilon(F) = 0$. The following, main result shows that generically the derivative of the pseudospectral abscissa with respect to ε exists and can be obtained as a by-product from its computation using Algorithm 1. There might be cases such that the derivative does not exist (for example a defective right-most point of multiplicity 2) but they have to be constructed in a non generic way.

Theorem 3 *Let $\hat{\varepsilon} > 0$ and assume that $\lambda_{\hat{\varepsilon}}$ is a globally rightmost point of $\Lambda_{\hat{\varepsilon}}$. Let $(u_{\hat{\varepsilon}}, v_{\hat{\varepsilon}})$ be a pair of (left, right) singular vectors corresponding to $\sigma_n(\sum_{i=0}^{m} A_i p_i(\lambda_{\hat{\varepsilon}}))$. Let Assumption 2 be satisfied, with the triple $(\lambda_{\hat{\varepsilon}}, u_{\hat{\varepsilon}}, v_{\hat{\varepsilon}})$ solving (18)-(21) for $\varepsilon = \hat{\varepsilon}$. There exists a constant $\delta > 0$ and a continuous function*

$$(\hat{\varepsilon} - \delta, \ \hat{\varepsilon} + \delta) \ni \varepsilon \mapsto (\lambda(\varepsilon), u(\varepsilon), v(\varepsilon))$$

such that $(\lambda(\varepsilon), u(\varepsilon), v(\varepsilon))$ satisfies (18)-(21) for all $\varepsilon \in (\hat{\varepsilon} - \delta, \ \hat{\varepsilon} + \delta)$, such that $(\lambda(\hat{\varepsilon}), u(\hat{\varepsilon}), v(\hat{\varepsilon})) = (\lambda_{\hat{\varepsilon}}, u_{\hat{\varepsilon}}, v_{\hat{\varepsilon}})$, as well as

$$\mathrm{Re}\left(\lambda'(\varepsilon)\right) = \frac{\sum\limits_{i=0}^{m} \frac{|p_i(\lambda_{\varepsilon})|}{w_i}}{u_{\varepsilon}^* \left(\sum\limits_{i=0}^{m} A_i \, p_i'(\lambda_{\varepsilon})\right) v_{\varepsilon} - \varepsilon \, \sum_{i=0}^{m} \left(\frac{\bar{p}_i(\lambda_{\varepsilon}) p_i'(\lambda_{\varepsilon})}{w_i |p_i(\lambda_{\varepsilon})|}\right)}, \tag{26}$$

for all $\varepsilon \in (\hat{\varepsilon} - \delta, \ \hat{\varepsilon} + \delta)$. Moreover, if the globally rightmost point is unique[1] then we have

$$\frac{d\alpha_{\varepsilon}}{d\varepsilon}\bigg|_{\varepsilon = \hat{\varepsilon}} = Re\left(\lambda'(\hat{\varepsilon})\right). \tag{27}$$

The proof can be found in [18]. Note that the differentiability of λ in a neighbourhood of $\hat{\varepsilon}$ follows from the implicit function theorem applied to (18)-(21).

It follows from Theorem 3 that the pseudospectral abscissa function is differentiable whenever the globally rightmost point of the pseudospectrum is unique and the technical Assumption 2 is satisfied. This is the case for almost all ε, that is, except, e.g., for special choices where the pseudospectrum exhibits for instance more than one globally rightmost point in the upper half complex plane. The fact that the derivative must be strictly positive can be readily seen from expression (26) and optimality condition (21). Finally, formulae (26)-(27) allow us to apply Newton's method to solve

$$\alpha_{\varepsilon}(F) = 0$$

for ε, in order to compute the distance to instability, that is

$$\varepsilon_{k+1} = \varepsilon_k - \frac{\alpha_{\varepsilon_k}(F)}{\frac{d\alpha_{\varepsilon}}{d\varepsilon}(F)\bigg|_{\varepsilon = \varepsilon_k}}, \qquad k = 0, 1, \ldots \tag{28}$$

In some cases it may occur that the distance to instability is not attained at a unique point and the ε-pseudospectrum is tangential to the imaginary axis at multiple points. In this situation the pseudospectral abscissa function is in general not differentiable. Newton's method may run into trouble, in the sense that the rightmost point of the pseudospectrum, computed in the inner iteration, may start alternating between approximations of the different globally rightmost points. Such a situation, which is non-generic unless the problem data result already from an optimization of the distance over parameters, can easily be detected from the output of the algorithm.

---

[1] As we assume that the spectrum is symmetric w.r.t. the real axis, only the eigenvalues in the upper half plane are considered.

4.2 The novel method: mixing the inner and outer iteration loop

Newton's iteration (28) is characterized by two nested loops. In the outer iteration loop, the value of ε is updated. Intrinsic to Newton's method, the convergence is quadratic in the generic case. In an inner iteration loop, the pseudospectral abscissa is computed for a fixed value of ε using the linearly converging Algorithm 1. In order to decrease the computational cost, one may argue that an accurate computation of the pseudospectral abscissa is not needed in the first outer iterations, but only when ε is close to $\mu(F)$. One can go a step further and replace the computation of the pseudospectral abscissa by one iteration step of Algorithm 1, and update ε based on the current value of λ_k, as if it had already converged to the globally rightmost point. This idea lies at the basis of Algorithm 2. We call the underlying method the *method of adaptive perturbations* because in every iteration step both the value of ε and the perturbations on the coefficient matrices are updated. Note that a fixed point $(\lambda, \varepsilon, u, v)$ of the iteration satisfies the conditions (18)-(21), and in addition we have $\text{Re}(\lambda) = 0$.

It should be noted that in case condition (30) and (31) do not allow a unique solution, the same remedies explained for Algorithm 1 apply. This situation has, however, not occurred in our experiments.

Algorithm 2 Method of adaptive perturbations

Initialize $(\varepsilon_0, \lambda_0, u_0, v_0)$ and repeat for $k = 1, 2, \ldots$

1. Compute λ_k as the rightmost eigenvalue of

$$\left(\sum_{i=0}^{m} (A_i + \delta A_i) p_i(\lambda) \right) v = 0, \qquad (29)$$

   where

$$\delta A_i = -\frac{\overline{p_i(\lambda_{k-1})}}{w_i |p_i(\lambda_{k-1})|} u_{k-1} v_{k-1}^* \varepsilon_{k-1}, \qquad 0 \le i \le m.$$

2. Define u_k and v_k as the left and right eigenvectors of (29) associated with λ_k, which are scaled such that

$$u_k^* u_k = v_k^* v_k = 1, \qquad c^* u_k > 0, \qquad (30)$$

$$u_k^* \left(\sum_{i=0}^{m} A_i p_i'(\lambda_k) \right) v_k - \varepsilon_k \sum_{i=0}^{m} \frac{\overline{p_i(\lambda_k)} p_i'(\lambda_k)}{w_i |p_i(\lambda_k)|} > 0. \qquad (31)$$

3. Update ε according to

$$\varepsilon_k = \varepsilon_{k-1} - \frac{\text{Re}(\lambda_k) \sum_{i=0}^{m} \frac{|p_i(\lambda_k)|}{w_i}}{u_k^* \left(\sum_{i=0}^{m} A_i p_i'(\lambda_k) \right) v_k - \varepsilon_{k-1} \sum_{i=0}^{m} \left(\frac{\bar{p}_i(\lambda_k) p_i'(\lambda_k)}{w_i |p_i(\lambda_k)|} \right)}$$

In all our experiments, represented in Section 5, Algorithm 2 turns out to be significantly faster then the iteration (28), while loss of convergence due to a nesting of iterations has not been observed. We note that, also with the update of ε, the method

only relies on the computation of the rightmost eigenvalue of perturbed problems, characterized by rank-one updates on the original system matrices. The latter property can be exploited by iterative solvers, both in matrix vector products and in solving linear systems, where the Sherman-Morrison-Woodbury formula plays a key-role. The overall cost of Algorithm 2 is comparable to the cost of Algorithm 1.

Algorithm 2 provides a fast method for computing the distance to instability, yet it is a local method, whose convergence to the correct solution relies on the choice of starting values. This is inherited from the fact that Algorithm 1 is based on a characterization of locally rightmost points of the ε-pseudospectrum. A stationary point of the iteration in Algorithm 2 might correspond to the situation where a locally rightmost point of the pseudospectrum is on the imaginary axis, while the distance to instability corresponds to the value of ε for which the globally rightmost point of the pseudospectrum is on the imaginary axis. One way to prevent such a situation and to improve the reliability is to run the algorithm multiple times, initiated with several dominant eigenvalues and corresponding eigenvectors. Here the term dominant corresponds to having a large real part and / or a large eigenvalue condition number (see the discussion in [17]), which are combined in the criterion

$$\left| \frac{u^* \left(\sum_{i=0}^{m} A_i p_i'(\lambda) \right) v}{\operatorname{Re}(\lambda)} \right|,$$

where (λ, u, v) is the eigentriple under consideration. Another solution is to generate starting values with the global method discussed in the next section.

5 Distance to instability via global optimization

From Proposition 2 the following expression for the distance to instability is obtained:

$$\mu(F) = \inf_{\omega \in \mathbb{R}} f(\omega), \tag{32}$$

where

$$f(\omega) := \sigma_n \left(\sum_{i=0}^{m} A_i p_i(j\omega) \right) \left[\sum_{i=0}^{m} \frac{|p_i(j\omega)|}{w_i} \right]^{-1},$$

i.e., f is the global mimimum of (frequency depending) scaled singular value curves of the parameterized matrix

$$\sum_{i=0}^{m} A_i p_i(j\omega). \tag{33}$$

Note that the objective function f is smooth almost everywhere when polynomial or exponential functions $\{p_i\}_{i=0}^{m}$ are used. This enables the recently proposed optimization method of [14], suitable for finding the global minimum of a minimum eigenvalue function of a parametrized Hermitian matrix. In order to find a global minimum of f in an interval $[\omega^l, \omega^r]$, in the ℓ-th iteration this method computes the global minimum on this interval of a function of the form

$$\max_{1 \le k \le \ell-1} f_k(\omega) \tag{34}$$

where the function f_k, $k \in \{1,\ldots,\ell-1\}$ is a (lower) support function of f at $\omega = \omega_k$, i.e., a function satisfying $f_k(\omega_k) = f(\omega_k)$ and $f_k(\omega) \leq f(\omega)$, $\forall \omega \in [\omega^l, \omega^r]$. Denoting the minimizer of function (34) by ω_ℓ, the whole process repeats itself with f_ℓ added to the set of support functions.

In [14] a detailed proof of convergence is presented, which relies on the property that the functions f_k provide global lower bounds on f. Because the function f may have discontinuities in the derivative due to the intersection of singular value curves of (33), nonsmooth (e.g. piecewise quadratic) support functions are needed to guarantee convergence, at the price of a complex subproblem of minimizing (34). In [14] also a more practical algorithm is presented and available for download, which relies on simple functions of the form

$$f_k(\omega) = f(\omega_k) + f'(\omega_k)(\omega - \omega_k) - \gamma(\omega - \omega_k)^2, \tag{35}$$

where $-\gamma$ is a lower bound on the second derivative of the singular value curves under investigation. In case the smallest singular value curve of (33) does not intersect another singular curve in the interval $[\omega^l, \omega^r]$, convergence of the algorithm to the global minimum is guaranteed, as all functions (35) are lower support functions for f. In the other case, convergence is not guaranteed, yet the algorithm performs very well in practice and proves to be very robust (see [14]), which may be explained by the property that a minimum eigenvalue function of a Hermitian matrix (in our case a minimum singular value function) is smooth in the neighborhood of a minimum. The core of our code for minimizing (32) is this optimization algorithm of [14]. Other important components are:

1. A routine which, for a given value of ω, returns the objective function and its derivative whenever it is differentiable. The computation of f mainly amounts to computing the smallest singular value of matrix $F(j\omega)$. Hence, a specific solver for nonlinear eigenvalue problems is not required, in contrast to the methods proposed in the previous section. In the generic case where $p_i(j\omega) \neq 0, i = 0,\ldots m$ and, in addition, the smallest singular value of $F(j\omega)$ is simple, the derivative $f'(\omega)$ exists and we can express

$$f'(\omega) = -\frac{\mathrm{Im}\left(u^* \left(\sum_{i=0}^m A_i p_i'(j\omega)\right) v\right)}{\sum_{i=0}^m \frac{|p_i(j\omega)|}{w_i}} + \frac{\sigma_n \left(\sum_{i=0}^m A_i p_i(j\omega)\right) \sum_{i=0}^m \frac{\mathrm{Im}\left(p_i(-j\omega) p_i'(j\omega)\right)}{w_i |p_i(j\omega)|}}{\left(\sum_{i=0}^m \frac{|p_i(j\omega)|}{w_i}\right)^2},$$

where u and v are normalized left and right singular vectors corresponding to the smallest singular value of $F(j\omega)$.

2. Prior knowledge of a compact interval which contains the global minimizer over $\mathbb{R}$. For the delay eigenvalue problem such an interval can be computed as in [20], but the obtained bounds may be conservative. Almost always the heuristic choice $[0, \omega_m]$ is sufficient, where $\omega_m = 1.2\, \mathrm{Im}(\lambda_m)$, with λ_m the dominant eigenvalue with largest imaginary part. A practical choice is to consider three dominant eigenvalues.

3. A lower bound $-\gamma$ on the second derivative of the scaled singular value curves, holding over the whole interval under consideration. It is impractical to compute

the second derivative and therefore, in our implementation, a piece-wise cubic approximating function of f over the interval $[0, \omega_m]$ is used to estimate a bound.

The choice of γ is the critical factor in the application of the method of [14], as analytical expressions for lower bounds on $f''(\omega)$ are –to the best of our knowledge– not available, even not for the standard eigenvalue problem. On the one hand, if the estimated value of γ is too small, there is a risk of no convergence to the global minimum. On the other hand, choosing a safe but conservative bound, leads to slow convergence. As a consequence a large number of function evaluations are needed if a high accuracy is requested. A robust and efficient algorithm can be obtained by combining the global method with the local one presented in the previous section: first the global method is run with a "safe" value of γ, but only until a rough approximation of the global optimum is obtained. Subsequently, the approximation is used to initialize Algorithm 2, which, on its turn, allows to refine the results up to the desired precision. In the next Section we will call this two-step procedure the hybrid algorithm.

6 Numerical experiments

We illustrate the methods for computing the distance to instability of Section 4 and Section 5. We compare the MATLAB implementations of the Newton iteration (28), the method with adaptive perturbations (Algorithm 2), and the global optimization algorithm for linear eigenvalue problems, quadratic eigenvalue problems, and for a small- and large-scale delay eigenvalue problem.

For computing the distance to instability of a nonlinear eigenvalue problem with one of the root finding methods (Newton's iteration (28) and Algorithm 2), we need a solver for computing the rightmost eigenvalue λ_{RM}. In the recent literature [10,24, 12,5], there exist several general nonlinear eigenvalue solvers, which are not only applicable to small problems but also to large sparse problems. In these numerical experiments the quadratic eigenvalue problem is first linearized and then λ_{RM} can be easily computed using `eig` or `eigs`. For computing λ_{RM} of the delay problems we used the delay eigenvalue solver proposed in [11]. On the other hand, the method based on global optimization (Section 5) for computing the distance to instability of a nonlinear eigenvalue problem only requires the computation of the smallest singular value of a constant matrix. For this we used `svd` or `svds`.

6.1 Linear eigenvalue problems

For the experiments on linear eigenvalue problems we take two examples (`'airy'` and `'transient'`) of the EIGTOOL collection [25] and one example (`'lshape'`) of the Harwell–Boeing collection [4]. This last example is shifted with shift $\sigma = 7$ in order to stabilize the matrix. The results of the computations of the distance to instability for these linear eigenvalue problems are displayed in Table 1.

The `'airy'` example shows the performance gain of the adaptive perturbation and optimization method compared to Newton's iteration (28). However, for the

'transient' example the gain of the optimization method is lost because of a too large estimate for the lower bound on the second derivative. The execution time of the Newton iteration and the method with adaptive perturbations for the 'lshape' example is similar because the start value ε_0 for both methods is zero, resulting after one iteration in $\varepsilon_1 = -(u_0^* v_0) \mathrm{Re}\,(\lambda_0)$ and this happens to be the distance to instability. Therefore the number of iterations of both methods is limited.

Table 1 Linear eigenvalue problems [4,25]: distance to instability ε_d and corresponding pseudospectral abscissa α_{ε_d} computed with Newton's method, the adaptive perturbation method, and the optimization method. The last two columns display the required number of iterations and the computation time, respectively.

Problem	Method	ε_d	α_{ε_d}	# iter	Time
airy	Newton	4.814833246948 e-02	2.92 e-11	22	392.1 s
($n = 99$)	Adaptive	4.814833245230 e-02	6.42 e-12	114	1.77 s
	Optimization	4.814833244747 e-02	7.44 e-15	127	0.87 s
transient	Newton	5.402011219865 e-10	-5.75 e-09	10	1.02 s
($n = 100$)	Adaptive	5.402091562084 e-10	-9.88 e-08	10	0.24 s
	Optimization	5.402008418433 e-10	-1.42 e-08	34874	197.3 s
lshape	Newton	9.108293989259 e-03	2.33 e-15	1	3.27 s
($n = 3466$, sparse)	Adaptive	9.108293989257 e-03	6.21 e-16	3	3.17 s
	Optimization	9.108293989353 e-03	9.67 e-14	2	0.19 s

6.2 Polynomial eigenvalue problems

For the experiments on polynomial eigenvalue problems we take three (shifted) examples ('hospital', 'pdde_stability' and 'sign2') of the NLEVP collection [1]. We applied perturbations to all system matrices and took unity weights, i.e., $w_i = 1$. The results of the computations are displayed in Table 2.

The 'hospital' problem shows that the root finding methods (Newton's iteration (28) and Algorithm 2) converge to a locally rightmost point, whereas the method based on optimization (Section 5) converges to the globally rightmost point. However, to achieve full accuracy of the global minimum, the optimization method requires a lot of iterations, since for this problem we needed to take to into account ten dominant eigenvalues. Therefore, a wise combination of the proposed Algorithms 2 and the global optimization method can tackle this limitation.

In particular, first execute the optimization method with high tolerance. If the lower bound is good, this can be done very fast. Assume the minimizer $\tilde{\omega}$ is found. Then the point $j\tilde{\omega}$ lies on the boundary of the ε-pseudospectrum where ε equals the objective function value. The corresponding perturbation can be constructed from the singular vectors by using [9, Lemma 1.1] for linear problems and [17, Proposition 3.1] for nonlinear problems. Subsequently, adaptive perturbations can be constructed by starting from $\lambda_0 = j\tilde{\omega}$, and ε equal to the objective function value and the aforementioned perturbation. This almost always results in a global solution with high accuracy, see, e.g., Table 2 where we set the required tolerance to 1e-12.

Table 2 Quadratic eigenvalue problems [1]: distance to instability ε_d and corresponding pseudospectral abscissa α_{ε_d} computed with Newton's method, the adaptive perturbation method, the optimization method, and a hybrid version of the previous two methods. The last two columns display the required number of iterations and the computation time, respectively.

Problem	Method	ε_d	α_{ε_d}	# iter	Time
hospital ($n=24$)	Newton	8.139884576182 e-02	0.58	3	0.09 s
	Adaptive	8.139884576180 e-02	0.58	8	0.05 s
	Optimization	4.400202122331 e-02	1.33 e-14	1067	25.00 s
	Hybrid	4.400202122334 e-02	5.37 e-13	803/8	14.66 s
pdde_stability ($n=225$, sparse)	Newton	9.231159788143 e-01	-3.38 e-14	5	78.88 s
	Adaptive	9.231159788137 e-01	-2.94 e-13	35	40.55 s
	Optimization	9.231159788142 e-01	-6.55 e-14	44	1.16 s
	Hybrid	9.231159788148 e-01	3.40 e-13	13/23	30.22 s
sign2 ($n=81$)	Newton	6.686158451176 e-01	4.54 e-14	13	552.10 s
	Adaptive	6.686158451149 e-01	8.50 e-13	467	44.46 s
	Optimization	6.686158450798 e-01	-5.81 e-11	88	0.66 s
	Hybrid	6.686158450920 e-01	8.36 e-13	20/67	6.50 s

6.3 Delay eigenvalue problems

The first delay example is the following small-scale delay eigenvalue problem

$$\left(\lambda I - A_0 - A_1 e^{-\lambda \tau} \right) x = 0, \tag{36}$$

where $\tau = 1$ and

$$A_0 = \begin{bmatrix} -5 & 1 \\ 2 & -6 \end{bmatrix}, \qquad A_1 = \begin{bmatrix} -2 & 1 \\ 3 & -1 \end{bmatrix}.$$

We assume that A_0 and A_1 are perturbed and choose unity weights, i.e., $w_0 = w_1 = 1$. The corresponding results are displayed in Table 3. We remark that Newton's method requires much less iterations than the adaptive perturbation method for the same accuracy. However, the computation cost for a Newton iteration is much higher. Note also that the computation time of the global optimization method may vary and depends on the estimate of γ. Therefore, the hybrid method is a good compromise for global convergence, accuracy and computation time.

The second delay example is a large-scale problem coming from a PDE with delay [11],

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} + a_0(x) v(x,t) + a_1(x) v(\pi - x, t - 1),$$

where $a_0(x) = -2\sin(x)$, $a_1(x) = 2\sin(x)$, and $v_x(0,t) = v_x(\pi,t) = 0$. The second derivatives in space are approximated with central differences. This gives rise to a standard delay eigenvalue problem of the form (36), with one delay and sparse matrices A_0 and A_1. The number of spatial discretization points is taken such that $n = 5000$. We take the weights $w_0 = w_1 = 1/2$. The corresponding results for the distance to instability are also displayed in Table 3. Again, we obtained similar results as for the small-scale delay problem. Note also that the difference in computation cost between Newton's method and the adaptive perturbation method increases with the problem size n.

Table 3 Delay eigenvalue problems: distance to instability ε_d and corresponding pseudospectral abscissa α_{ε_d} computed with Newton's method, the adaptive perturbation method, the optimization method, and a hybrid version of the previous two methods. The last two columns display the required number of iterations and the computation time, respectively.

Problem	Method	ε_d	α_{ε_d}	# iter	Time
small-scale ($n = 2$)	Newton	1.762769038185791	$-7.46\,\mathrm{e}\text{-}16$	5	4.79 s
	Adaptive	1.762769038190436	$8.86\,\mathrm{e}\text{-}13$	37	0.80 s
	Optimization	1.762769038184915	$-3.21\,\mathrm{e}\text{-}13$	53	0.15 s
	Hybrid	1.762769038189088	$6.73\,\mathrm{e}\text{-}13$	26/35	0.78 s
large-scale ($n = 5000$, sparse)	Newton	0.249999991774308	$4.58\,\mathrm{e}\text{-}15$	5	75.93 s
	Adaptive	0.249999991774340	$4.19\,\mathrm{e}\text{-}14$	8	6.60 s
	Optimization	0.249999991776921	$3.27\,\mathrm{e}\text{-}12$	30	18.68 s
	Hybrid	0.249999991774678	$4.60\,\mathrm{e}\text{-}13$	15/5	13.57 s

7 Concluding remarks

Two algorithms have been adapted, combined and implemented for computing the distance to instability, for both linear and nonlinear eigenvalue problems. The two algorithms are well suited for large-scale sparse problems. Although for the method with adaptive perturbations only convergence to a locally rightmost point of the pseudospectrum on the imaginary axis can be guaranteed, a high accuracy can be achieved. The second method directly solves an optimization problem inferred from the characterization (12). As the main advantage, the global optimum can be found, yet the convergence rate can be slow if the second derivative of the objective function cannot be accurately estimated. Therefore, a hybrid algorithm is recommended which almost always converges globally with high accuracy and a reasonable computation time.

Recently, several algorithms have been proposed for computing extremal points of *structured* and / or *real* pseudospectra of matrices, for the Euclidean and the Frobenius norm (see, e.g., [8] for real pseudospectra). This class of methods is based on either a discrete iteration or on a differential equation on a manifold of low rank matrices, which originate from the property that in the cases under consideration the boundary of the pseudospectra can be reached by applying low rank perturbations to the matrix. It is expected that the idea of updating ε, behind Algorithm 2, applies to all these methods, which, in this way, could be extended to the computation of the corresponding distances to instability. Extending the presented approach based on global optimization to structured or real perturbations seems more difficult. For the standard eigenvalue problem, an extension to real perturbations measured with the Euclidean norm is still possible. The key is to replace the expression for the distance to instability (12) by a characterization in terms of real structured singular values [13, 21]. However, computing real structured singular values is significantly more demanding computationally than computing (standard) singular values.

Acknowledgments

References

1. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: A Collection of Nonlinear Eigenvalue Problems. MIMS EPrint 2011.116, Manchester Institute for Mathematical Sciences, The University of Manchester, Manchester (2011)
2. Burke, J.V., Lewis, A.S., Overton, M.L.: Robust stability and a crisscross algorithm for pseudospectra. IMA J. Numer. Anal. **23**(3), 359–375 (2003)
3. Byers, R.: A Bisection Method for Measuring the Distance of a Stable Matrix to the Unstable Matrices. SIAM J. Sci. Statist. Comput. **9**(5), 875–881 (1988)
4. Duff, I.S., Grimes, R.G., Lewis, J.G.: Users' guide for the harwell-boeing sparse matrix collection (release i) (1992)
5. Effenberger, C.: Robust successive computation of eigenpairs for nonlinear eigenvalue problems. Tech. Rep. 27, Ecole Polytechnique Federale de Lausanne, Lausanne (2012)
6. Freitag, M.A., Spence, A.: A Newton-based method for the calculation of the distance to instability. Linear Algebra Appl. **435**(12), 3189–3205 (2011)
7. Guglielmi, N., Lubich, C.: Differential Equations for Roaming Pseudospectra: Paths to Extremal Points and Boundary Tracking. SIAM J. Numer. Anal. **49**(3), 1194–1209 (2011)
8. Guglielmi, N., Lubich, C.: Low-Rank Dynamics for Computing Extremal Points of Real Pseudospectra. SIAM J. Matrix Anal. Appl. **34**(1), 40–66 (2013)
9. Guglielmi, N., Overton, M.L.: Fast Algorithms for the Approximation of the Pseudospectral Abscissa and Pseudospectral Radius of a Matrix. SIAM J. Matrix Anal. Appl. **32**(4), 1166–1192 (2011)
10. Güttel, S., Van Beeumen, R., Meerbergen, K., Michiels, W.: NLEIGS: A class of robust fully rational Krylov methods for nonlinear eigenvalue problems. Tech. Rep. TW633, Department of Computer Science, KU Leuven, Leuven (2013). URL `http://www.cs.kuleuven.be/publicaties/rapporten/tw/TW633.abs.html`
11. Jarlebring, E., Meerbergen, K., Michiels, W.: A Krylov Method for the Delay Eigenvalue Problem. SIAM J. Sci. Comput. **32**(6), 3278–3300 (2010)
12. Jarlebring, E., Michiels, W., Meerbergen, K.: A linear eigenvalue algorithm for the nonlinear eigenvalue problem. Numer. Math. **122**(1), 169–195 (2012)
13. Karow, M., Kokiopoulou, E., Kressner, D.: On the computation of structured singular values and pseudospectra. Systems Control Lett. **59**(2), 122–129 (2010)
14. Kılıç, M., Mengi, E., Yıldırım, E.A.: Numerical optimization of eigenvalues of Hermitian matrix functions. Tech. Rep. arXiv:1109.2080, Koç University, Instanbul (2011)
15. Kressner, D.: Finding the distance to instability of a large sparse matrix. In: Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE, vol. 31–35. IEEE (2006)
16. Michiels, W., Green, K., Wagenknecht, T., Niculescu, S.I.: Pseudospectra and stability radii for analytic matrix functions with application to time-delay systems. Linear Algebra Appl. **418**(1), 315–335 (2006)
17. Michiels, W., Guglielmi, N.: An Iterative Method for Computing the Pseudospectral Abscissa for a Class of Nonlinear Eigenvalue Problems. SIAM J. Sci. Comput. **34**(4), A2366–A2393 (2012)
18. Michiels, W., Guglielmi, N.: Computing the distance to instability for large-scale nonlinear eigenvalue problems. In: Proceedings of the European Control Conference. Zürich, Switzerland (2013)
19. Michiels, W., Niculescu, S.I.: Stability and stabilization of time-delay systems. An eigenvalue based approach. SIAM (2007)
20. Michiels, W., Unal, H.U.: Evaluating and approximating fir filters: an approach based on functions of matrices. Submitted (2013)
21. Qiu, L., Bernhardsson, B., Rantzer, A., Davison, E., Young, P., Doyle, J.: A formula for computation of the real stability radius. Automatica **31**(6), 879–890 (1995)
22. Tisseur, F., Higham, N.J.: Structured Pseudospectra for Polynomial Eigenvalue Problems, with Applications. SIAM J. Matrix Anal. Appl. **23**(1), 187–208 (2001)
23. Trefethen, L.N., Embree, M.: Spectra and Pseudospectra: the behavior of non-normal matrices and operators. Princeton University Press (2005)
24. Van Beeumen, R., Meerbergen, K., Michiels, W.: A Rational Krylov Method Based on Hermite Interpolation for Nonlinear Eigenvalue Problems. SIAM J. Sci. Comput. **35**(1), A327–A350 (2013)
25. Wright, T.G.: Eigtool: A graphical tool for nonsymmetric eigenproblems (2002). URL `http://www.comlab.ox.ac.uk/pseudospectra/eigtool`