Measures for Robust Stability and Controllability

by

Emre Mengi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
September 2006

Michael L. Overton

We have not known a single great scientist who could not discourse freely and interestingly with a child.

John Steinbeck, the Log From the Sea of Cortez.

To my parents and grandparents

Preface

The degree of robustness of a stable and controllable dynamical system is closely related to the sensitivity of the eigenvalues. Ill-conditioned eigenvalues are partly responsible for the nonrobustness of a stable or controllable system. However, they should not necessarily ring the alarm bell. This thesis investigates some of the possible quantities indicating the degree of robustness of a dynamical system; all intrinsically have ties with the sensitivity of the eigenvalues. Apart from the intuitive motivation that a dynamical system is surrounded by uncertainties and we would like to know how much uncertainty is tolerable, most of these quantities have proved to be useful at other contexts. Thanks to the Kreiss matrix theorem, the robust stability measures discussed in this work give insight into the transient behavior of the dynamical system. In a recent work by Braman, Byers and Mathias, the distance to uncontrollability that we elaborate on is shown to measure the convergence of the QR iteration to particular eigenvalues and experimentally this discovery is verified to provide significant speed-ups to the traditional implementations of the QR algorihm.

The thesis focuses on computation of these quantities rather than further analyzing their effectiveness in indicating the robustness. The algorithms devised benefit from the equivalent singular-value or eigenvalue optimization characterizations. The difficulty that is tried to overcome is the nonconvex nature of these optimization problems. Surprisingly the global minima of these nonconvex problems are located reliably and with a decent amount of work (usually by means of a fast converging algorithm with a cubic cost at each iteration) by the help of structured eigenvalue solvers and iterative eigenvalue solvers.

For the completion of this work I am indebted to my advisor Michael Overton for various reasons. I could not be involved in these fascinating problems that connect optimization, numerical linear algebra and control theory without him being the advisor. Secondly almost every person researching on numerical linear algebra or numerical optimization that I gained the acquaintance is due to him. As further elaborated below suggestions by some of these people were influential in some of the better parts of this thesis. Finally the completion of a Ph.D. thesis requires one to keep his or her motivation up which is inevitably affected by the

support of the advisor. I have received the sincerest support from Michael Overton.

It is also due to Michael Overton that I got to know Adrian Lewis whose works I have always admired. Some of the results in this thesis are simple modifications of the results of Adrian Lewis for analogous problems. During this work I have spent two summers at the technical university of Berlin thanks to Volker Mehrmann who hosted me generously and who made me realize the importance of the structure-preserving eigenvalue solvers. During my first visit I had the opportunity to interact with Daniel Kressner from whom I received valuable comments on the algorithms for the distance to uncontrollability. The earlier attempts to solve some of the problems in this thesis were made by Ralph Byers during the late 1980s and early 1990s. His works inspired the algorithms in this thesis. Nick Trefethen and Mark Embree have been the strong advocates of the pseudospectra for various applications and, as far as this work is concerned, in understanding the transient behavior of dynamical systems. I am grateful to their detailed comments regarding the algorithm for the pseudospectral radius. Ming Gu, who suggested the first polynomial time algorithm for the distance to uncontrollability, initiated the idea of extracting the real eigenvalues efficiently in the computation of the distance to uncontrollability. The high-precision algorithm for the distance to uncontrollability was a product of interacting with Ming Gu and his students Jianlin Xia and Jiang Zhu. I am also grateful to Olof Widlund and Margaret Wright for accepting duties in the committees at various stages of this thesis. Their feedback improved the quality of the thesis greatly.

Abstract

A linear time-invariant dynamical system is robustly stable if the system and all of its nearby systems in a neighborhood of interest are stable. An important property of robustly stable systems is that they decay asymptotically without exhibiting significant transient behavior. The first part of this thesis work focuses on measures revealing the degree of robust stability. We put special emphasis on pseudospectral measures, those based on the eigenvalues of nearby matrices for a first-order system or matrix polynomials for a higher-order system. We present new algorithms with quadratic rate of convergence for the computation of pseudospectral measures and analyze their accuracy in the presence of rounding errors. We also provide an efficient new algorithm for computing the numerical radius of a matrix, the modulus of the outermost point in the set of Rayleigh quotients of the matrix.

We call a system robustly controllable if it is controllable and remains controllable under perturbations of interest. We describe efficient methods for the computation of the distance to the closest uncontrollable system. Our first algorithm for the distance to uncontrollability of a first-order system depends on a grid and is well-suited for low-precision approximation. We then discuss algorithms for high-precision approximation. These are based on the bisection method of Gu and the trisection variant of Burke-Lewis-Overton. These algorithms require the extraction of the real eigenvalues of matrices of size $O(n^2)$, typically at a cost of $O(n^6)$, where n is the dimension of the state space. We propose a new divide-and-conquer algorithm for real eigenvalue extraction that reduces the cost to $O(n^4)$ on average in both theory and practice, and is $O(n^5)$ in the worst case. For higher-order systems we derive a singular-value characterization and exploit this characterization for the computation of the higher-order distance to uncontrollability to low precision. The algorithms in this thesis assume that arbitrary complex perturbations are applicable and require the extraction of the imaginary eigenvalues of Hamiltonian matrices (or even matrix polynomials) or the unit eigenvalues of symplectic pencils (or palindromic matrix polynomials). MATLAB implementations of all algorithms discussed are freely available.

Keywords: dynamical system, stability, controllability, pseudospectrum, pseudospectral abscissa, pseudospectral radius, field of values, numerical radius, distance to instability, distance to uncontrollability, real eigenvalue extraction, Arnoldi, inverse iteration, eigenvalue optimization, polynomial eigenvalue problem

Contents

Chapter 1

Motivation and Background

1.1 Continuous-time control systems

We consider the linear time-invariant dynamical system

$$K_k x^{(k)}(t) + K_{k-1} x^{(k-1)}(t) + \cdots + K_0 x(t) = Bu(t), \tag{1.1a}$$

$$y(t) = Cx(t) + Du(t) \tag{1.1b}$$

with the initial conditions

$$x(0) = c_0,\ x'(0) = c_1,\ \ldots, x^{(k-1)}(0) = c_{k-1}$$

where $u(t) : \mathbb{R} \to \mathbb{C}^m$ is the input control, $y(t) : \mathbb{R} \to \mathbb{C}^p$ is the output measurement, $x(t) : \mathbb{R} \to \mathbb{C}^n$ is the state function and $K_0, K_1, \ldots K_k \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, $C \in \mathbb{C}^{p\times n}$, $D \in \mathbb{C}^{p\times m}$ are the coefficient matrices. We assume that the leading coefficient K_k is nonsingular. Equation (1.1) is called a *state-space* description of the dynamical system.

It is desirable that a dynamical system possess certain properties. Below we briefly review stability, controllability, observability and stabilizability. Some of these properties concern the *autonomous* state-space system when $u(t) = 0$. The associated matrix polynomial

$$P(\lambda) = \sum_{j=0}^{k} \lambda^j K_j$$

and its eigenvalues play crucial roles in the equivalent characterizations of these properties. The scalar $\lambda' \in \mathbb{C}$ is an *eigenvalue* of the polynomial P if $\det P(\lambda') = 0$ in which case the left null space of $P(\lambda')$ is called the *left eigenspace*, the right null space of $P(\lambda')$ is called the *right eigenspace*, each vector in the left eigenspace is called a *left eigenvector* and each vector in the right

eigenspace is called a *right eigenvector* corresponding to λ'. For a given matrix $A \in \mathbb{C}^{n\times n}$, when $P(\lambda) = A - \lambda I$, we obtain the definition of the standard eigenvalue problem. In this case the characteristic polynomial $c_s(\lambda) = \det(P(\lambda))$ is of degree n, therefore has exactly n roots or equivalently A has n eigenvalues. In general the degree of the polynomial $c(\lambda) = \det(P(\lambda))$ is equal to nk, when the leading coefficient is nonsingular and is strictly less than nk otherwise. Let the degree of the characteristic polynomial be l, then the matrix polynomial P has $nk - l$ infinite eigenvalues and l finite eigenvalues. Specifically when $k = 1$, the linear matrix function $\lambda K_1 + K_0$ is called a *pencil* and each λ such that the pencil $\lambda K_1 + K_0$ is rank deficient is called a *generalized eigenvalue*.

Stability: The autonomous state-space system is *stable* if for all initial conditions the state vector vanishes asymptotically, that is

$$\lim_{t\to\infty} \|x(t)\| = 0, \ \forall c_0, c_1, \ldots, c_{k-1}.$$

The stability of the system (1.1) is equivalent to the condition that all of the eigenvalues of P lie in the open left half of the complex plane. Furthermore the real part of the rightmost eigenvalue, called the *spectral abscissa* of P,

$$\alpha(P) = \max\{\text{Re } \lambda : \lambda \in \mathbb{C} \ \text{ s.t. } \ \det \mathrm{P}(\lambda) = 0\}$$

determines the decay rate as for all x_0 and for all $\delta > 0$

$$\|x(t)\| = O(e^{t(\alpha(P)+\delta)}).$$

Controllability: The state-space system (1.1) or the tuple $(K_0, K_1, \ldots, K_k, B)$ is called *controllable* if it is possible to drive the system into any desired state at any given time by the proper selection of the input. Controllability is equivalent to the rank problem [36] (a generalization of the characterization for the first-order system when $k = 1$ and $K_1 = I$ due to Kalman [42])

$$\text{rank } [P(\lambda) \ \ B] = n, \ \ \forall \lambda \in \mathbb{C}. \tag{1.2}$$

Let an uncontrollable systems with zero initial conditions be the mapping $y = f_1(u)$. The uncontrollable system is not minimal in the sense that there exists a system $y = f_2(u)$ whose state lies in a smaller space and for all u the equality $f_1(u) = f_2(u)$ holds.

Observability: The autonomous state-space system is *observable* if each possible output is caused by a unique initial condition. While controllability is

defined in terms of the set of possible output for a given initial state, observability is defined in the reverse direction in terms of the set of initial conditions for a given output. It is not surprising that the rank characterization

$$\text{rank} \begin{bmatrix} P(\lambda) \\ C \end{bmatrix} = n, \quad \forall \lambda \in \mathbb{C} \tag{1.3}$$

for observability is analogous to (1.2).

Stabilizability: The state-space system is *stabilizable* if for all initial conditions there exists an input so that the state vector decays asymptotically. When the system is controllable, the system is stabilizable, but the reverse implication is not necessarily true. The stabilizability of the state-space system can be reduced to the condition

$$\text{rank} \, [P(\lambda) \;\; B] = n, \; \forall \lambda \in \mathbb{C}_+,$$

which is same as the controllability characterization except that the rank tests need to be performed only in the closed right half of the complex plane.

An important special case of (1.1) is the first-order system

$$x'(t) = Ax(t) + Bu(t), \quad x(0) = c_0 \tag{1.4a}$$

$$y(t) = Cx(t) + Du(t) \tag{1.4b}$$

with $A \in \mathbb{C}^{n \times n}$. All the previous discussion applies for the first-order system. Characterizations of stability, controllability, observability and stabilizability in this case can be obtained by making the substitution $P(\lambda) = A - \lambda I$. For the controllability of the first-order system another equivalent characterization is that the controllability matrix

$$[B \;\; AB \;\; A^2B \;\; \ldots \;\; A^{n-1}B] \tag{1.5}$$

has full row rank. For a detailed description of all these fundamental properties and their characterizations for the first-order system we refer to the book [22]. The characterizations for the higher-order system are derived from the corresponding characterizations for the first-order system using a linearization procedure which is a common way of embedding a higher-order system into a first-order system. In the paper [58], the equivalent characterization for the higher-order distance to uncontrollability was proved.

1.2 Discrete-time control systems

In many applications the dependence on time is discrete. In this case the state-space system maps a discrete input function to a discrete output function in the form

$$K_k x_{j+k} + K_{k-1} x_{j+k-1} + \cdots + K_0 x_j = B u_{j+k}, \tag{1.6a}$$

$$y_j = C x_j + D u_j \tag{1.6b}$$

with the initial conditions

$$x_0 = c_0, \ldots, x_{k-1} = c_{k-1}.$$

The definitions of stability, controllability, observability and stabilizability are similar to those for continuous-time systems. For discrete-time systems the moduli of the eigenvalues of the associated matrix polynomial P are relevant rather than the real parts of the eigenvalues. In particular the system is stable if and only if all of the eigenvalues of the polynomial P are strictly inside the unit circle, and the *spectral radius* of P

$$\rho(P) = \max\{|\lambda| : \lambda \in \mathbb{C} \text{ s.t. } \det \mathrm{P}(\lambda) = 0\}$$

determines the asymptotic decay rate. The equivalent characterizations of controllability and observability are identical to the continuous case, while for stabilizability the rank tests need to be performed on and outside the unit circle, that is the discrete system is stabilizable if the condition

$$\operatorname{rank} [P(\lambda) \;\; B] = n, \; \forall \lambda \text{ s.t. } |\lambda| \geq 1$$

holds.

1.3 Robustness

The state-space system is usually an approximation of a complicated system that is subject to uncertainty. Therefore it is desirable that the properties described in the previous section are preserved when the system is changed by small perturbations.

Consider the autonomous first-order system with the coefficient matrix A equal to the 50×50 "Grcar" matrix, a Toeplitz matrix with -1 on the sub-diagonal and diagonal, 1 on the first, second and third super-diagonals and all of the other entries zero. The system is stable, since the spectral abscissa is equal to -0.3257. However perturbations ΔA with norm on the order of 10^{-2}

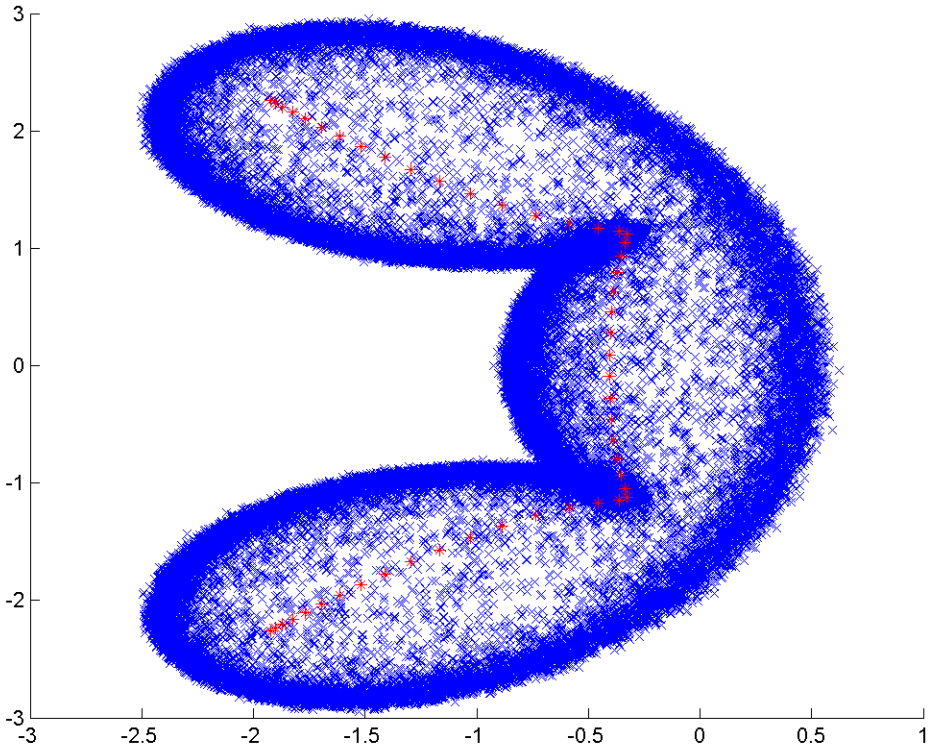

Figure 1.1: Eigenvalues of the matrices obtained by applying normally distributed perturbations with mean zero and standard deviation 10^{-2} to the Grcar matrix. The blue crosses and red asterisks illustrate the eigenvalues of the perturbed matrices and the Grcar matrix, respectively.

are sufficient to move some of the eigenvalues to the right-half plane significantly away from the imaginary axis. In Figure 1.1 the eigenvalues of 1000 nearby matrices are shown. The nearby matrices are obtained by perturbing with matrices whose entries (both the real parts and the imaginary parts) are chosen mutually independently from a normal distribution with zero mean and standard deviation 10^{-2}. The real part of the rightmost eigenvalue among the 1000 matrices selected is 0.6236. Therefore the system is not robustly stable against perturbations on the order of 10^{-2}.

The first-order state-space system with

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.32 & 0.3 \\ 0.2 & 0.5 & 0.26 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 500 \\ 500 \\ 500 \end{bmatrix} \tag{1.7}$$

is controllable. Indeed the smallest singular value of the controllability matrix (1.5) is 1.23. However, the perturbation

$$\Delta A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.02 & 0 \\ 0 & 0 & 0.04 \end{bmatrix}$$

to A yields an uncontrollable system. Figure 1.2 illustrates the variation in the minimum singular values of the controllability matrix as the second and third diagonal entries of A are perturbed. The minimum singular value for a particular perturbation can be determined from the color bar on the right. The perturbations to the second diagonal entry of A affect the minimum singular value of the controllability matrix more drastically.

1.3.1 Robust stability

We call an autonomous control system *robustly stable* if it is stable and all of the systems in a given neighborhood are stable. Formally the system (1.1) with $u(t) = 0$ is robustly stable with respect to given sets $\Delta_0, \Delta_1, \ldots, \Delta_k$ if the perturbed system

$$\sum_{j=0}^{k} (K_j + \Delta K_j) x^{(j)}(t) = 0$$

is stable for all $\Delta K_j \in \Delta_j,\ j = 0, \ldots, k$. Robust stability for the discrete system (1.6) is defined analogously. In general the neighborhood of interest depends on the application. For instance, if the coefficient matrices in (1.1) or in (1.6) are real, it may be desirable to allow only real perturbations, that is the sets $\Delta_j,\ j = 0, \ldots, k$, contain only real matrices. In other applications the

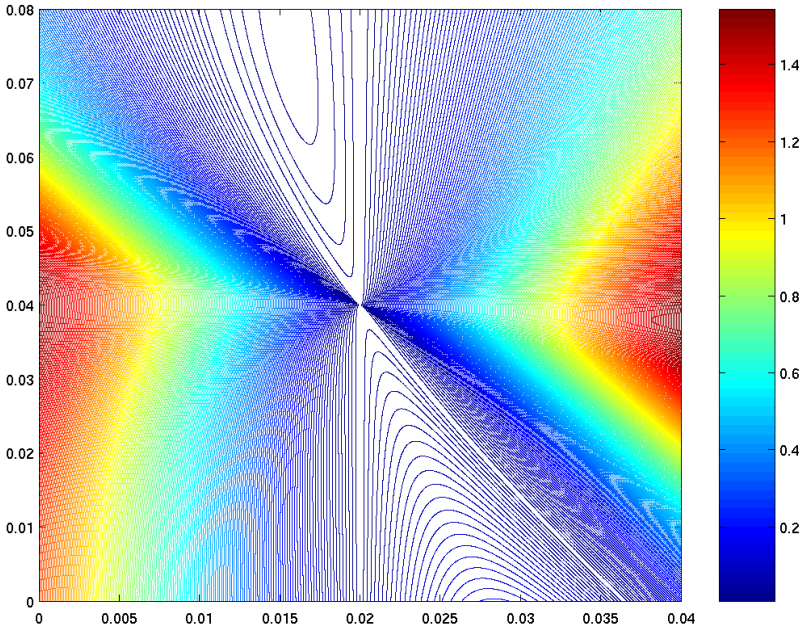

Figure 1.2: The level sets of the minimum singular value of the controllability matrix for the pair $(A + \Delta A, B)$ where (A, B) is defined by (1.7) and the perturbation matrix ΔA has nonzero entries only on the second and the third diagonal. The horizontal and vertical axis correspond to negative perturbations to the second and positive perturbations to the third diagonal entries, respectively.

perturbations may have multiplicative structure, *e.g.* the set $\Delta_j,\ j = 0, \ldots, k$ consists of matrices of the form $F_j \delta_j G_j$ where $F_j \in \mathbb{C}^{n \times l_j}$, $G_j \in \mathbb{C}^{r_j \times n}$ are fixed and $\delta_j \in \mathbb{C}^{l_j \times r_j}$ is a variable. For real or structured perturbations [74], [65], [43], [37], [62], [28] and [66] are good resources. In this thesis we consider only complex unstructured perturbations. The next two chapters are mainly devoted to robust stability measures based on the pseudospectrum, the set of eigenvalues of nearby matrices or polynomials, and the field of values, the set of Rayleigh quotients of a given matrix. Particular emphasis is put on the computation of such measures.

The primary reason for interest in robust stability measures is to gain information about the level of robustness. Additionally, all of these measures are relevant to the magnitude of the transient behavior of the system. As discussed in §1.1 and §1.2, asymptotic behaviors of the continuous system (1.1) and the discrete system (1.6) are completely determined by the spectral abscissa and the spectral radius of the associated polynomial, respectively. Recall the Grcar matrix whose rightmost real eigenvalue is in the left half-plane and considerably away from the imaginary axis. Figure 1.3 shows that for the Grcar matrix the norm of the exponential e^{At} decays fast (at $t = 40$ the norm drops to the order of 10^{-4}) but only after exhibiting a transient peak around $t = 10$ where the norm is on the order of 10^3. On the other hand Figure 1.4 shows that for the upper triangular matrix with all of the upper triangular and diagonal entries equal to -0.3 the norm of e^{At} decays monotonically. One might see the sensitivity of the eigenvalues of the Grcar matrix (as illustrated by Figure 1.1) as responsible for the initial peak. While, as we will see, there is some truth in this observation, the sensitivity of the eigenvalues alone does not explain why the upper triangular matrix has initial behavior consistent with its asymptotic behavior. The upper triangular matrix is similar to the Jordan block of size 50 with the diagonal entries equal to -0.3, so the eigenvalue -0.3 is defective and extremely ill-conditioned. Indeed in Figure 1.5 the norm of the powers of the same matrix with spectral radius equal to 0.3 reaches the order of 10^6. The measures in Chapter 2 and Chapter 3 explain why the discrete system with the upper triangular coefficient matrix exhibits an initial growth unlike the continuous system. (Also see [70] for more examples whose initial behaviors can be understood by the help of the pseudospectra.)

Below we review the basic tools that we will depend upon in the chapters on robust stability measures.

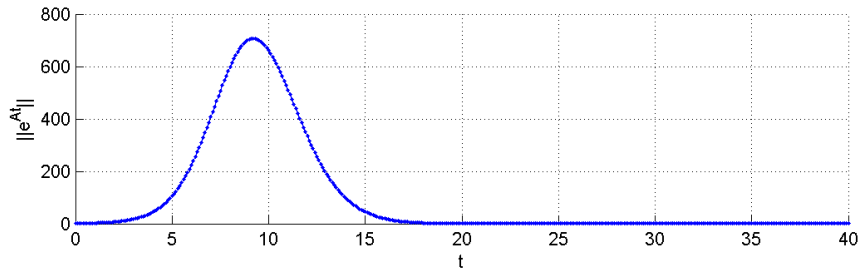

Figure 1.3: The norm of the exponential e^{At} on the vertical axis as a function of time on the horizontal axis for the Grcar matrix.

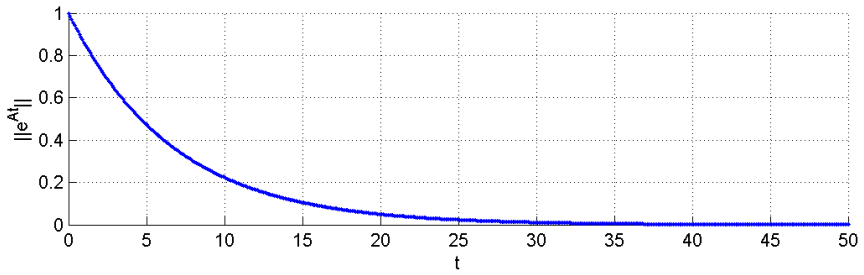

Figure 1.4: The norm of the exponential e^{At} for an upper triangular matrix with all of the upper triangular and diagonal entries equal to -0.3.

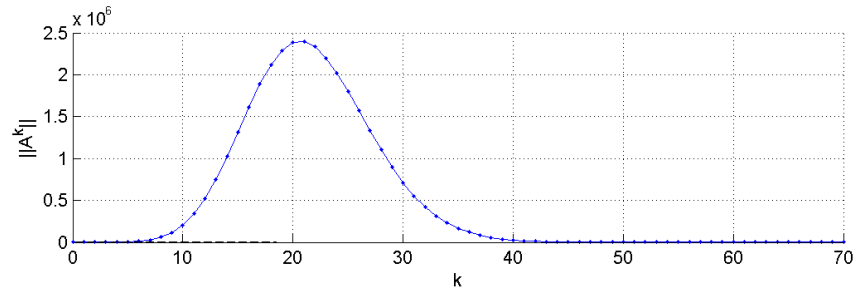

Figure 1.5: The norm of the powers A^j for the upper triangular matrix with all of the upper triangular and diagonal entries equal to -0.3.

Distance to instability

One of the measures of robust stability is the *distance to instability* which we define as

$$\beta(P,\gamma) = \inf\{\|[\Delta K_k \ \Delta K_{k-1} \ \ldots \ \Delta K_0]\| : \\ (K_k + \gamma_k \Delta K_k, K_{k-1} + \gamma_{k-1}\Delta K_{k-1}, \ldots, \gamma_0 \Delta K_0) \text{ is unstable}\} \tag{1.8}$$

where the vector $\gamma = [\gamma_k \ \gamma_{k-1} \ \ldots \gamma_0]$ consists of nonnegative scalars not all zero. Above and throughout this thesis $\|\cdot\|$ denotes the 2-norm unless otherwise stated. Our motivation in introducing the vector γ is mainly to restrict the perturbations to some combination of coefficient matrices by setting all of the other γ_j to zero. It also serves the purpose of scaling the perturbations to the coefficients. For instance, one may be interested in perturbations in a relative sense with respect to the norm of the coefficients in which case it is desirable to set $\gamma = [\|K_k\| \ \ldots \ \|K_1\| \ \|K_0\|]$. Let $\mathbb{C}_g$ and $\mathbb{C}_b$ partition the complex plane with the open set $\mathbb{C}_g$ denoting the stable region and $\mathbb{C}_b$ denoting the unstable region. Thus for continuous systems $\mathbb{C}_g$ is the open left half-plane and for discrete systems $\mathbb{C}_g$ is the open unit disk. Using the continuity of eigenvalues with respect to perturbations to the coefficient matrices, the definition of the distance to instability can be restated as

$$\beta(P,\gamma) = \inf_{\lambda \in \partial\mathbb{C}_b} \nu(\lambda, P, \gamma) \tag{1.9}$$

with $\partial\mathbb{C}_b$ denoting the boundary of $\mathbb{C}_b$ and

$$\nu(\lambda, P, \gamma) = \inf\{\|\Delta K_k \ \ldots \ \Delta K_0\| : \\ \det(P + \Delta P)(\lambda) = 0 \text{ where } \Delta P(\lambda) = \sum_{j=0}^{k} \lambda^j \gamma_j \Delta K_j\}.$$

The latter optimization problem is a structured singular value problem [65] and therefore the equation (1.9) can be simplified to (see [28] for the details)

$$\beta(P,\gamma) = \inf_{\lambda \in \partial\mathbb{C}_b} \sigma_{\min}\left[\frac{P(\lambda)}{p_\gamma(|\lambda|)}\right] \tag{1.10}$$

where

$$p_\gamma(x) = \sqrt{\sum_{j=0}^{k} \gamma_j^2 x^{2j}}. \tag{1.11}$$

For the first-order system with $k = 1$, $K_0 = A$, $K_1 = -I$ and $\gamma = [0\ 1]$ the definition (1.8) reduces to the one introduced by Van Loan [54],

$$\beta(A) = \inf\{\|\Delta A\| : A + \Delta A \text{ is unstable}\} \tag{1.12}$$

and the characterization (1.10) reduces to

$$\beta(A) = \inf_{\lambda \in \partial\mathbb{C}_b} \sigma_{\min}[A - \lambda I]. \tag{1.13}$$

Pseudospectrum

For a given positive real scalar ϵ and a vector of nonnegative scalars $\gamma = [\gamma_k, \ldots, \gamma_0]$ (see the remark following (1.8) for the motivation in introducing γ), we define the *ϵ-pseudospectrum* of P as the set

$$\Lambda_\epsilon(P, \gamma) = \{z \in \Lambda(P + \Delta P) : \\ \Delta P(\lambda) = \sum_{j=0}^{k} \gamma_j \lambda^j \Delta K_j \text{ where } \|\Delta K_k \ \ldots \ \Delta K_0\| \le \epsilon,\ j = 0, \ldots, k\}. \tag{1.14}$$

In [67] the equivalent singular value characterization

$$\Lambda_\epsilon(P, \gamma) = \{\lambda \in \mathbb{C} : \sigma_{\min}[P(\lambda)] \le \epsilon p_\gamma(|\lambda|)\} \tag{1.15}$$

is given where p_γ was defined in (1.11) . The ϵ-pseudospectrum of a matrix

$$\Lambda_\epsilon(A) = \{z \in \Lambda(A + \Delta A) : \|\Delta A\| \le \epsilon\} \tag{1.16}$$

has been extensively studied by Trefethen [68, 69, 70] and can be efficiently computed by means of the characterization

$$\Lambda_\epsilon(A) = \{\lambda \in \mathbf{C} : \sigma_{\min}(A - \lambda I) \le \epsilon\}. \tag{1.17}$$

Note that (1.16) is obtained from (1.14) by setting $\gamma_0 = 1, \gamma_1 = 0$ so that $K_1 = -I$ is not subject to perturbation.

The MATLAB package *EigTool* [73] is a free toolbox for the computation of the ϵ-pseudospectrum of a matrix. For techniques for the computation of the ϵ-pseudospectrum of a matrix and matrix polynomial, see [69] and [67], respectively.

Field of values

The field of values of the matrix A consists of the Rayleigh quotients of A, *i.e.*

$$F(A) = \{z^*Az : z \in \mathbb{C}^n,\ \|z\| = 1\}. \tag{1.18}$$

It is a convex set containing the eigenvalues of A. When A is normal, the field of values is the convex hull of the eigenvalues of A. Furthermore when A is Hermitian, the field of values lies on the real axis. In general the projections of $F(A)$ onto the real and imaginary axis are the fields of values of the Hermitian part $F(H(A))$ and the skew-Hermitian part $F(N(A))$ respectively, where $H(A) = \frac{A+A^*}{2}$ and $N(A) = \frac{A-A^*}{2}$. For the derivation of these and other geometric properties of $F(A)$ see [39].

For the matrix polynomial P the field of values can be generalized to

$$F(P) = \{\lambda : \exists z \in \mathbb{C}^n \text{ s.t. } z^*P(\lambda)z = 0\}$$

which simplifies to (1.18) for the matrix A when $P(\lambda) = A - \lambda I$. The paper [52] and the book [29] analyze the geometrical properties of the field of values of a matrix polynomial.

1.3.2 Robust controllability

The system (1.1) or (1.6) is called *robustly controllable* if the system itself is controllable as are all the nearby systems in a given neighborhood. More specifically, robust controllability with respect to given sets $\Delta_0, \Delta_1, \ldots, \Delta_k, \Delta_B$ implies that the perturbed system

$$\sum_{j=0}^{k} (K_j + \Delta K_j)x^{(j)}(t) = (B + \Delta B)u(t)$$

is controllable for all $\Delta K_j \in \Delta_j,\ j = 0, \ldots, k$ and $\Delta B \in \Delta_B$. As with stability the perturbations may be constrained to be real (see [40]) or to have structure, but in this thesis we focus on robust controllability with respect to unstructured complex perturbations.

One measure for the degree of robust controllability is the *distance to uncontrollability* which we define as

$$\begin{aligned}\tau(P, B, \gamma) = \inf\{\|[\Delta K_k \ \ \ldots \ \ \Delta K_1 \ \ \Delta K_0 \ \ \Delta B]\| :\\ (K_k + \gamma_k \Delta K_k, \ldots, K_0 + \gamma_0 \Delta K_0, B + \Delta B) \text{ is uncontrollable}\}\end{aligned} \tag{1.19}$$

where the vector $\gamma = [\gamma_k \ \ldots \ \gamma_1 \ \gamma_0]$ fulfills the scaling task as in the definition of the distance to instability in (1.8). When $k = 1$, $K_1 = I$, $K_0 = -A$ and

$\gamma = [0\ 1]$ we recover the definition introduced by Paige [61] for the first-order system

$$\tau(A, B) = \inf\{\|\Delta A \ \ \Delta B\| : \text{the pair } (A + \Delta A, B + \Delta B) \text{ is uncontrollable}\}. \tag{1.20}$$

Note that even though in this thesis we consider the distances in the 2-norm, we will see that the definitions (1.19) and (1.20) in the 2-norm and Frobenius norm are equivalent. We present various techniques for computing the distance to uncontrollability for the first-order system (1.4) and for the higher-order system (1.1) in Chapter 4 and Chapter 5, respectively. The distance functions for observability and stabilizability can be defined similarly. All of the algorithms for the distance to uncontrollability can be modified in a straightforward fashion for the distances to unobservability and unstabilizability.

1.4 Outline and contributions

This thesis is organized as follows. We start with a review of some of the robust stability measures for continuous systems in Chapter 2 with emphasis on pseudospectral measures. Specifically we review the algorithm by Burke, Lewis and Overton [12] for the ϵ-pseudospectral abscissa of a matrix, the real part of the rightmost point in the ϵ-pseudospectrum, show that it is backward stable under reasonable assumptions and extend it to matrix polynomials. In Chapter 3 on robust stability of discrete systems, we introduce an analogous algorithm for the ϵ-pseudospectral radius, the modulus of the outermost point in the ϵ-pseudospectrum, analyze the algorithm and generalize it for matrix polynomials. In the same chapter we also present an algorithm for computing the numerical radius of a matrix, the modulus of the outermost point in the field of values, that combines the ideas in [9] and [34]. Chapter 4 is devoted to the computation of the distance to uncontrollability for first-order systems. We focus on two algorithms for the first-order distance to uncontrollability. The first one works on a grid and is suitable for low-precision approximation. The second one reduces the computational cost of the algorithms in [31] and [13] to $O(n^4)$ from $O(n^6)$. When comparing the computational costs of the algorithms, throughout this thesis we assume that the computation of the eigenvalues of a matrix or pencil of size n is an atomic operation with a cost of $O(n^3)$ unless otherwise stated. Computation of the distance to uncontrollability for higher-order systems is addressed in Chapter 5. We derive a singular value characterization and describe an algorithm for low precision exploiting the singular value characterization. We illustrate the performance of the algorithms in practice with the numerical examples at the end of each chapter. All of the numerical

experiments are performed with MATLAB 6.5 running on a PC with 1000 Mhz Intel processor and 256MB RAM.

The contributions of this thesis are summarized below.

- *Chapter 2:* A backward stability analysis is provided for the pseudospectral abscissa algorithm in [12], and the algorithm is extended to matrix polynomials.

- *Chapter 3:* An algorithm for the computation of the pseudospectral radius is presented. For generic cases the algorithm converges quadratically and is proved to be backward stable under reasonable assumptions. Also an algorithm for computing the numerical radius is described. To our knowledge both of the algorithms are the most efficient and accurate techniques available.

- *Chapter 4:* A low-precision approximation and a high-precision approximation technique for the first-order distance to uncontrollability are suggested. The low-precision approximation is usually more efficient than the other algorithms in the same category, particularly the algorithms in [16]. The high-precision technique reduces the computational cost of [31] and [13] from $O(n^6)$ to $O(n^4)$, both in theory and in practice. It uses a divide-and-conquer algorithm devised for real eigenvalue extraction which is potentially applicable to other problems.

- *Chapter 5:* A minimum singular value characterization for the higher-order distance to uncontrollability is given and an algorithm based on this characterization is introduced.

- *Software:* All of the algorithms in this thesis have been implemented in MATLAB and the software is freely available at [58].

Chapter 2

Robust Stability Measures for Continuous Systems

The robust stability and initial behavior of the continuous-time autonomous first-order system

$$x'(t) = Ax(t) \tag{2.1}$$

and the autonomous higher-order system

$$K_k x^{(k)}(t) + K_{k-1} x^{(k-1)}(t) + \cdots + K_0 x(t) = 0 \tag{2.2}$$

are the subject of this chapter.

Pseudospectra are known to be good indicators of robust stability. In particular, the real part of the rightmost point in the ϵ-pseudospectrum of A, called the ϵ-pseudospectral abscissa, is useful in determining the transient peaks and the degree of robustness of the stability of (2.1). In §2.1 we recall the quadratically convergent algorithm by Burke, Lewis and Overton [12] for the ϵ-pseudospectral abscissa of A, analyze its backward error and extend it to the higher-order system (2.2). In §2.2 computation of the distance to instability for (2.1) and (2.2) using the Boyd-Balakrishnan algorithm [9] is reviewed.

2.1 Pseudospectral abscissa

The largest of the real parts of the points in the ϵ-pseudospectrum of A is called the *ϵ-pseudospectral abscissa* of A,

$$\alpha_\epsilon(A) = \max\{\mathrm{Re}\, z : z \in \Lambda_\epsilon(A)\} \tag{2.3}$$

or equivalently

$$\alpha_\epsilon(A) = \max\{\mathrm{Re}\, z : \sigma_{\min}(A - zI) \le \epsilon\}. \tag{2.4}$$

Keeping in mind that the equality $\sup_{x_0} \|x(t)\| = \|e^{At}\|$ holds for all t, an immediate implication of the Kreiss matrix theorem [44]

$$\sup_{\epsilon>0} \frac{\alpha_\epsilon(A)}{\epsilon} \leq \sup_t \|e^{At}\| \leq en \sup_{\epsilon>0} \frac{\alpha_\epsilon(A)}{\epsilon}. \tag{2.5}$$

justifies the importance of the ϵ-pseudospectral abscissa in determining the magnitude of the maximum peak of the first-order autonomous system.

Figure 2.1 and Figure 2.2 illustrate the ϵ-pseudospectrum of the Grcar matrix and the upper triangular matrix whose initial behaviors are compared in §1.3.1. All of the pseudospectra plots in this thesis are generated by *EigTool* [73]. Both of the matrices have very sensitive eigenvalues; however, unlike the eigenvalues of the Grcar matrix, the perturbations to the upper triangular matrix move its eigenvalues towards the imaginary axis very little. With perturbations with norm $\epsilon = 10^{-3}$ applied to the Grcar matrix its eigenvalues cross the imaginary axis, indeed $\alpha_\epsilon = 0.13$. The lower bound in (2.5) implies that the norm of the matrix exponentials must exceed 130. On the other hand when perturbations with norm 10^{-3} are applied to the upper triangular matrix, it remains stable with $\alpha_\epsilon = -0.15$.

2.1.1 The algorithm of Burke, Lewis and Overton

To compute the pseudospectral abscissa, Burke, Lewis and Overton [12] introduced an algorithm with a quadratic rate of convergence for generic matrices. The algorithm was inspired by Byers' bisection algorithm [15] and its quadratically convergent variant by Boyd and Balakrishnan [9] for the distance to uncontrollability. It exploits the characterization (2.4) and is based on vertical and horizontal searches in the complex plane.

A *vertical search* finds the intersection points of a given vertical line with the ϵ-pseudospectrum boundary. The next result, proved in [12] (a simple generalization of a result of Byers [15]), implies that a vertical search can be achieved by solving an associated Hamiltonian eigenvalue problem (see §A.1 for the definition and properties of a Hamiltonian eigenvalue problem).

Theorem 1 (**Vertical Search**). *Let x be a real number and ϵ be a positive real number. The matrix $A-(x+yi)I$ has ϵ as one of its singular values if and only if the Hamiltonian matrix*

$$V(x,\epsilon) = \begin{bmatrix} xI - A^* & \epsilon I \\ -\epsilon I & A - xI \end{bmatrix} \tag{2.6}$$

has the imaginary eigenvalue yi.

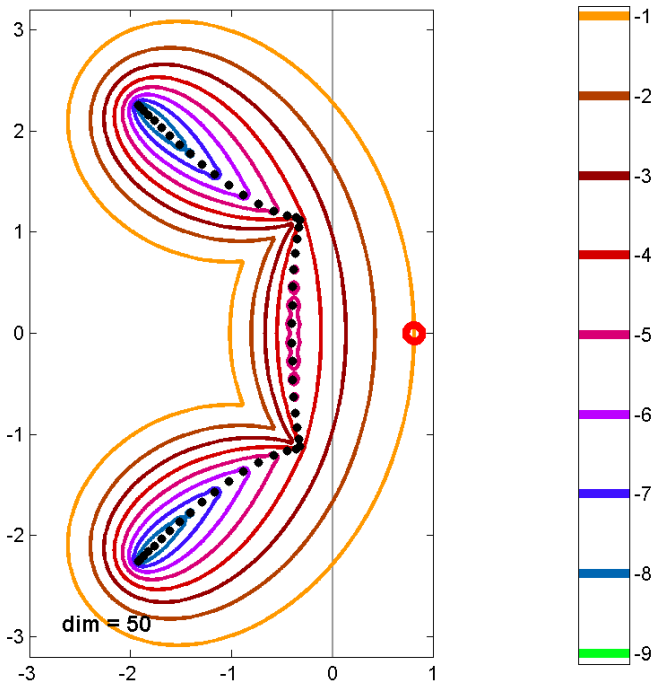

Figure 2.1: The boundary of the ϵ-pseudospectrum of the Grcar matrix is shown for various ϵ. The color bar on the right displays the value of ϵ in logarithmic scale corresponding to each color. The black dots and the red disk mark the locations of the eigenvalues and the location where the pseudospectral abscissa is attained for $\epsilon = 10^{-1}$, respectively. The gray line is the imaginary axis.

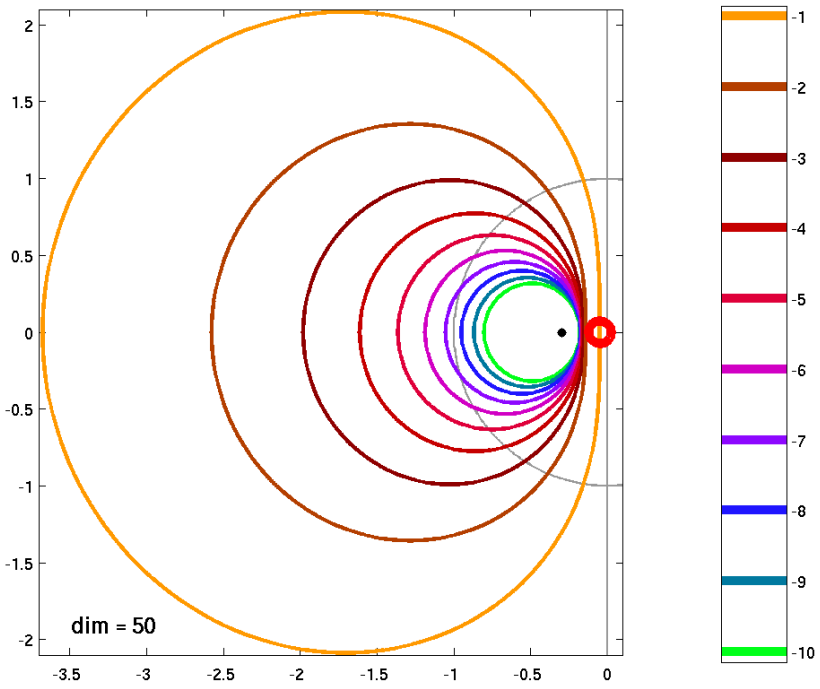

Figure 2.2: The boundary of the ϵ-pseudospectrum of the upper triangular matrix with the entries $a_{ij} = -0.3,\ j \geq i$ is displayed for various ϵ. The ϵ-pseudospectral abscissa for $\epsilon = 10^{-1}$ is attained at the location marked with the red disk. The gray arc is the part of the unit circle.

The intersection points of the vertical line at x can be found by computing the eigenvalues of $V(x,\epsilon)$ followed by a singular value test for each imaginary eigenvalue of $V(x,\epsilon)$. For each $yi \in \Lambda(V(x,\epsilon))$, we need to check whether $\sigma_{\min}(A-(x+yi)I) = \epsilon$. The current state of art for reliable and efficient computation of the imaginary eigenvalues of a Hamiltonian matrix is reviewed in §A.1.

In a *horizontal search* the aim is to determine the intersection point of a horizontal line and the ϵ-pseudospectrum boundary that is furthest to the right. The next theorem (see [12] for the proof) indicates that a horizontal search requires the solution of an associated Hamiltonian eigenvalue problem.

Theorem 2 (**Horizontal Search**). *Let y be a real number and ϵ be a positive real number. The largest x such that $A-(x+yi)I$ has ϵ as the smallest singular value is the imaginary part of the upper-most imaginary eigenvalue of*

$$\tilde{H}(y,\epsilon) = \begin{bmatrix} iA^* - yI & \epsilon I \\ -\epsilon I & iA + yI \end{bmatrix}. \tag{2.7}$$

To find the intersection point of the horizontal line at y and the ϵ-pseudospectrum boundary with the largest real part, it suffices to extract the largest imaginary eigenvalue of $\tilde{H}(y,\epsilon)$.

The criss-cross algorithm (Algorithm 1) for the ϵ-pseudospectral abscissa starts from the spectral abscissa as the initial estimate for the ϵ-pseudospectral abscissa. At each iteration a vertical search finds the intersection points of the vertical line passing through the current estimate and the ϵ-pseudospectrum boundary. From the intersection points it is easy to determine the segments of the vertical line lying inside the ϵ-pseudospectrum. From the midpoint of each segment a horizontal search is performed. The estimate for the pseudospectral abscissa is updated to the maximum value returned by the horizontal searches. Figure 2.3 displays how the algorithm proceeds on a 10×10 companion example that is taken from *EigTool*'s demo menu [73] and shifted by $-3.475I$, for $\epsilon = 10^{-4}$.

2.1.2 Backward error analysis

For the sake of simplicity we assume that $tol = 0$, and that we are given an implementation of Algorithm 1 in floating point arithmetic that produces a monotonically increasing sequence of estimates $\{\hat{x}^r\}$ and terminates when the vertical search fails to return an intersection point because of rounding errors. (The vertical search typically fails in practice when the estimate is sufficiently close to α_ϵ.)

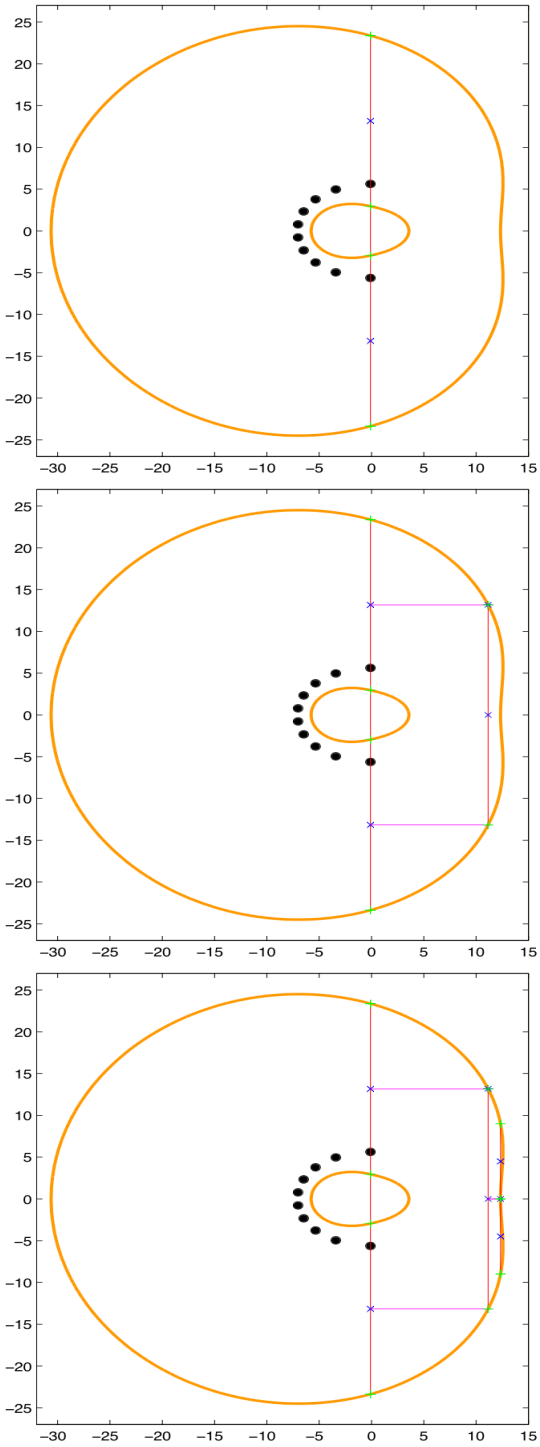

Figure 2.3: The progress of the criss-cross algorithm for the pseudospectral abscissa on a companion matrix example for $\epsilon = 10^{-4}$.

Algorithm 1 Criss-cross algorithm for the pseudospectral abscissa

Call:	$\hat{\alpha}_\epsilon \leftarrow$ `pspa`(A,ϵ,tol).
Input:	$A \in \mathbb{C}^{n\times n}$, $\epsilon \in \mathbb{R}_+$, $tol \in \mathbb{R}_+$ (tolerance for termination).
Output:	$\hat{\alpha}_\epsilon \in \mathbb{R}$, the estimate value for the ϵ-pseudospectral abscissa.

Let $x^0 = \alpha(A)$ and $j = 0$.
repeat
perform vertical search: Perform a vertical search to find the intersection points of the vertical line at x^j and the ϵ-pseudospectrum boundary. Using the intersection points, determine the segments I_1^j, I_2^j,..., $I_{m^j}^j$ on the vertical line at x^j lying inside the ϵ-pseudospectrum. Compute the set of midpoints of the segments

$$y^j = \{\frac{u_l^j + l_l^j}{2},\ \ l = 1, \ldots, m^j\},$$

where $I_l^j = (l_l^j, u_l^j)$, for $l = 1, \ldots, m^j$.
perform horizontal searches: Perform a horizontal search at each midpoint $y_l^j \in y^j$. Compute

$$x^{j+1} = \max\{x_\epsilon(y_l^j) : y_l^j \in y^j\} \tag{2.8}$$

where $x_\epsilon(y)$ is the result of the horizontal search at y.
increment j
until $x_j - x_{j-1} < tol$.
return x_j

Suppose the numerical algorithm terminates at the mth iteration and the value at termination is $\hat{x}^m$. We claim that provided a structure-preserving and backward stable Hamiltonian eigenvalue solver is used, the value returned, $\hat{x}^m$, is the $(\epsilon+\beta)$-pseudospectral abscissa of A, where $\beta = O(\delta_{\text{mach}}(\|A\|+\epsilon+\rho_\epsilon(A)))$, the constant δ_{mach} is the machine precision and $O(\delta_{\text{mach}})$ means "of the order of the machine precision" [71]. Note that this result does not depend on perturbations of the original matrix, but on the exact matrix for a perturbed value of ϵ. For backward error analysis it is sufficient to bound $\hat{x}^m$ from above and below in terms of the abscissa of nearby pseudospectra. That is, we aim to show that

$$\alpha_{\epsilon-\beta_l}(A) \leq \hat{x}^m \leq \alpha_{\epsilon+\beta_u}(A) \tag{2.9}$$

holds for some positive $\beta_l = O(\delta_{\text{mach}}(\|A\|+\epsilon+\rho_\epsilon(A)))$ and $\beta_u = O(\delta_{\text{mach}}(\|A\|+\epsilon+\rho_\epsilon(A)))$. Then the backward error of of the algorithm is $O(\delta_{\text{mach}}(\|A\|+\epsilon+\rho_\epsilon(A)))$ from the continuity of the ϵ-pseudospectral abscissa as a function of ϵ [11, 51, 50].

For deriving these bounds we need to consider the accuracy of the horizontal search and the vertical search. In a horizontal search, given a real y, we need to find the greatest x such that $\sigma_{\min}(A-(x+iy)I) = \epsilon$. In exact arithmetic the horizontal search for a given y is performed by extracting the greatest pure imaginary eigenvalue of the Hamiltonian matrix $\tilde{H}(y,\epsilon)$ defined by (2.7). In floating point arithmetic, when a structure preserving backward stable Hamiltonian eigenvalue solver as discussed in §A.1 is used, the horizontal search instead returns the imaginary part of the greatest pure imaginary eigenvalue of a perturbed Hamiltonian matrix

$$\tilde{L}(y,\epsilon) = \tilde{H}(y,\epsilon) + \tilde{E} \tag{2.10}$$

where $\|\tilde{E}\| = O(\delta_{\text{mach}}\|\tilde{H}(y,\epsilon)\|)$. Notice that $\|\tilde{H}(y,\epsilon)\| \leq 2(\|A\|+\epsilon+\rho_\epsilon(A))$ holds, so $\|\tilde{E}\| = O(\delta_{\text{mach}}(\|A\|+\epsilon+\rho_\epsilon(A)))$.

The estimate at termination, $\hat{x}^m$, is generated by a horizontal search at the previous iteration. Therefore for some y the perturbed Hamiltonian matrix $\tilde{L}(y,\epsilon)$ has $i\hat{x}^m$ as its greatest pure imaginary eigenvalue. According to the following theorem, having $i\hat{x}^m$ in the spectrum of $\tilde{L}(y,\epsilon)$ implies that $\hat{x}^m+iy$ belongs to a nearby pseudospectrum. We omit the proof because of its similarity to the proof of Theorem 18 in Chapter 3.

Theorem 3 (**Accuracy of the Horizontal Search**). *Suppose the Hamiltonian matrix $\tilde{L}(y,\epsilon)$ has the imaginary eigenvalue ix. Then $ix \in \Lambda(\tilde{H}(y,\epsilon+\beta))$ for some real β such that $|\beta| \leq \|\tilde{E}\|$.*

Now that we know the complex number $\hat{x}^m+iy$ belongs to the $(\epsilon+\beta)$-pseudospectrum for some $\beta \leq \|\tilde{E}\|$, we deduce the upper bound on $\hat{x}^m$,

$$\hat{x}^m \leq \alpha_{\epsilon+\|\tilde{E}\|}(A). \tag{2.11}$$

Recall that $\|\tilde{E}\| = O(\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A)))$ as desired.

To derive a lower bound we exploit the fact that the vertical search at the final iteration fails to return an intersection point. In a vertical search we are interested in the intersection points of the ϵ-pseudospectrum boundary and a given vertical line. This is achieved by computing the imaginary eigenvalues of the Hamiltonian matrix $V(x, \epsilon)$ defined by (2.6). The imaginary parts of the pure imaginary eigenvalues of $V(x, \epsilon)$ consist of a superset of the intersection points of the ϵ-pseudospectrum boundary with the vertical line through x. On the other hand, in floating point arithmetic, assuming a backward stable and structure-preserving algorithm is used, the potential intersection points we obtain are the imaginary parts of the imaginary eigenvalues of a Hamiltonian matrix

$$L(x, \epsilon) = V(x, \epsilon) + \hat{E} \tag{2.12}$$

with $\|\hat{E}\| = O(\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A)))$.

The termination of the algorithm at $x = \hat{x}^m$ occurs because the vertical search with $x = \hat{x}^m$ fails in floating point arithmetic, which in turn implies that the Hamiltonian matrix $L(\hat{x}^m, \epsilon)$ does not have any imaginary eigenvalue. Combining this fact with Theorem 4 below, we deduce that the vertical line at $\hat{x}^m$ does not intersect the boundary of nearby pseudospectra.

Theorem 4 (**Accuracy when the Vertical Search Fails**). *Suppose* $\epsilon > \|\hat{E}\|$. *If the matrix* $L(x, \epsilon)$ *does not have any pure imaginary eigenvalue, then for all* $\gamma \in (0, \epsilon - \|\hat{E}\|]$, *the matrix* $V(x, \gamma)$ *does not have any pure imaginary eigenvalue.*

Proof. Denote the eigenvalues of the Hermitian matrices $R(y) = JL(x, 0) - iyJ$ and $T(y) = JV(x, 0) - iyJ$ by $s_j(y)$ and $t_j(y)$, $j = 1 \dots 2n$, ordered in descending order. It follows from Weyl's Theorem [38][Theorem (4.3.1)] that the difference between the corresponding eigenvalues of $T(y)$ and $R(y)$ can be at most the norm of the perturbation matrix $\|\hat{E}\|$, *i.e* for all j and y,

$$|s_j(y) - t_j(y)| \leq \|\hat{E}\|. \tag{2.13}$$

Also notice that the eigenvalues of $T(y)$ are plus and minus the singular values of $A - (x + iy)I$. Moreover as $y \to \infty$, all of the singular values of $A - (x + iy)I$ are unbounded below, so using (2.13), we have

$$\lim_{y\to\infty} t_j(y) = \lim_{y\to\infty} s_j(y) = \begin{cases} \infty & \text{for } 1 \leq j \leq n, \\ -\infty & \text{for } n+1 \leq j \leq 2n. \end{cases} \tag{2.14}$$

Now since $L(x, \epsilon)$ does not have any imaginary eigenvalue, for all y

$$\det(JL(x, \epsilon) - iyJ) = \det(JL(x, 0) - iyJ - \epsilon I) = \det(R(y) - \epsilon I) \neq 0$$

is satisfied, meaning for all y and j, $s_j(y) \neq \epsilon$. For the sake of contradiction assume that for some real γ in the interval $(0, \epsilon - \|\hat{E}\|]$, the matrix $V(x,\gamma)$ has the imaginary eigenvalue $i\hat{y}$. Then

$$\det(JV(x,\gamma) - i\hat{y}J) = \det(JV(x,0) - i\hat{y}J - \gamma I) = \det(T(\hat{y}) - \gamma I) = 0.$$

In other words since γ is positive, there exists a $j \leq n$ such that $t_j(\hat{y}) = \gamma$. But according to (2.13), $s_j(\hat{y})$ and $t_j(\hat{y})$ cannot differ by more than $\|\hat{E}\|$. Hence we see that $s_j(\hat{y}) \leq \epsilon$. We infer from (2.14) and from the intermediate value theorem that there must be a $\tilde{y} \geq \hat{y}$ such that $s_j(\tilde{y}) = \epsilon$. Therefore we have a contradiction, completing the proof. □

Because of the assumption that the estimates are in increasing order, the computed value $\hat{x}^m$ is strictly greater than the spectral abscissa. Furthermore it cannot be between the spectral abscissa and $\alpha_{\epsilon-\|\hat{E}\|}$. Theorem 2.6 in [12] states that for all x values between the $(\epsilon - \|\hat{E}\|)$-pseudospectral abscissa and the spectral abscissa some part of the vertical line at x must lie inside the $(\epsilon - \|\hat{E}\|)$-pseudospectrum. But we infer from Theorem 4 that there is no y such that $\sigma_{\min}(A - (\hat{x}^m + iy)I) < \epsilon - \|\hat{E}\|$. Hence we conclude that the lower bound on $\hat{x}^m$

$$\alpha_{\epsilon-\|\hat{E}\|} \leq \hat{x}^m \tag{2.15}$$

is satisfied. Combined with (2.11) this completes the argument that a numerical implementation of Algorithm 1 generating an increasing sequence of estimates and terminating when the vertical search fails has a backward error on the order of $\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A))$.

2.1.3 The pseudospectral abscissa of a matrix polynomial

For the higher-order system the ϵ-pseudospectral abscissa is defined as

$$\alpha_\epsilon(P,\gamma) = \max\{\text{Re } z : z \in \Lambda_\epsilon(P,\gamma)\} \tag{2.16}$$

or

$$\alpha_\epsilon(P,\gamma) = \max\{\text{Re } z : \frac{\sigma_{\min}(P(z))}{p_\gamma(|z|)} \leq \epsilon\}. \tag{2.17}$$

Algorithm 1 easily generalizes for the higher-order system. All we need to explain is how to do vertical and horizontal searches on the polynomial ϵ-pseudospectrum.

The next theorem states how to perform vertical searches.

Theorem 5 (**Vertical Search on the Polynomial ϵ-Pseudospectrum**). *Given a real x and a positive real ϵ, let*

$$B_l(x) = \sum_{j=l}^{k} \left(\begin{array}{c} j \\ l \end{array} \right) x^{j-l} K_j$$

and

$$b_l(x) = (-1)^{l/2+1} \left(\sum_{j=l/2}^{k} \gamma_j^2 \left(\begin{array}{c} j \\ l/2 \end{array} \right) x^{2j-l} \right)$$

for $l = 1, \ldots, k$. At least one of the singular values of $\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}$ is equal to ϵ if and only if the matrix polynomial $\mathcal{V}(x,\epsilon) = \sum_{j=0}^{k} \lambda^j \mathcal{V}_j(x,\epsilon)$ has the eigenvalue yi where

$$\mathcal{V}_0(x,\epsilon) = \begin{bmatrix} b_0(x)\epsilon I & (B_0(x))^* \\ B_0(x) & -\epsilon I \end{bmatrix},$$

when l is odd,

$$\begin{aligned} \mathcal{V}_l(x,\epsilon) &= \begin{bmatrix} 0 & -(B_l(x))^* \\ B_l(x) & 0 \end{bmatrix} && 1 \le l \le k, \\ \mathcal{V}_l(x,\epsilon) &= 0 && k+1 \le l < 2k, \end{aligned}$$

and, when l is even,

$$\begin{aligned} \mathcal{V}_l(x,\epsilon) &= \begin{bmatrix} b_l(x)\epsilon I & (B_l(x))^* \\ B_l(x) & 0 \end{bmatrix} && 1 \le l \le k, \\ \mathcal{V}_l(x,\epsilon) &= \begin{bmatrix} b_l(x)\epsilon I & 0 \\ 0 & 0 \end{bmatrix} && k+1 \le l \le 2k. \end{aligned}$$

Proof. The matrix $\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}$ has ϵ as a singular value if and only if the matrix is

$$\begin{bmatrix} -\epsilon I & \frac{(P(x+yi))^*}{p_\gamma(\sqrt{x^2+y^2})} \\ \frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})} & -\epsilon I \end{bmatrix}$$

or, by multiplying the leftmost blocks and upper blocks by $p_\gamma(\sqrt{x^2+y^2})$, the matrix

$$\begin{bmatrix} -\epsilon(p_\gamma(\sqrt{x^2+y^2}))^2 I & P(x+yi)^* \\ P(x+yi) & -\epsilon I \end{bmatrix} = \sum_{l=0}^{2k} (iy)^l \mathcal{V}_l(x,\epsilon)$$

is singular, that is iy is an eigenvalue of the matrix polynomial $\mathcal{V}(x,\epsilon)$. □

To find the intersection points of the boundary of $\Lambda_\epsilon(P)$ and the vertical line at x, we first extract the imaginary eigenvalues of $\mathcal{V}(x,\epsilon)$. If yi is an imaginary eigenvalue of $\mathcal{V}(x,\epsilon)$, the above theorem ensures that the matrix $\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}$ has ϵ as a singular value, but not necessarily the smallest one. Therefore at a second step we check whether $\sigma_{\min}\left(\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}\right)$ for all $yi \in \Lambda(\mathcal{V}(x,\epsilon))$. Note that the even coefficients of $\mathcal{V}(x,\epsilon)$ are Hermitian, while its odd coefficients are skew-Hermitian. Such a matrix polynomial is called a $*$-even matrix polynomial in §A.3 and has eigenvalues either purely imaginary or in pairs $(\lambda, -\bar{\lambda})$. See §A.3 for further discussions on $*$-even matrix polynomials and particularly how to reliably extract the imaginary eigenvalues of such matrix polynomials.

The horizontal search can be accomplished by extracting the uppermost imaginary eigenvalue of a $*$-even polynomial. Unlike the ϵ-pseudospectrum of a matrix, the ϵ-pseudospectrum of a matrix polynomial is not bounded when $\gamma_k > 0$ and $\sigma_{\min}\left(\frac{K_k}{\gamma_k}\right) \leq \epsilon$. We shall only consider the case when the ϵ-pseudospectrum is bounded and therefore $\alpha_\epsilon(P)$ is finite.

Theorem 6 (**Horizontal Search on the Polynomial ϵ-Pseudospectrum**). *Assume that a real y and a positive real ϵ satisfying $\sigma_{\min}\left(\frac{K_k}{\gamma_k}\right) > \epsilon$ are given. Let*

$$C_l(y) = \sum_{j=l}^{k} \begin{pmatrix} j \\ l \end{pmatrix} (-y)^{j-l}(-i)^j K_j$$

and

$$c_l(y) = (-1)^{l/2+1}\left(\sum_{j=l/2}^{k} \gamma_j^2 \begin{pmatrix} j \\ l/2 \end{pmatrix} (-y)^{2j-l}\right)$$

for $l = 1, \ldots, k$. The largest x such that $\sigma_{\min}\left(\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}\right) = \epsilon$ is the imaginary part of the uppermost imaginary eigenvalue of the matrix polynomial $\tilde{\mathcal{H}}(y,\epsilon) = \sum_{l=0}^{k} \lambda^j \tilde{\mathcal{H}}_l(y,\epsilon)$ with

$$\tilde{\mathcal{H}}_0(y,\epsilon) = \begin{bmatrix} -\epsilon\gamma_0^2 I & K_0^* \\ K_0 & -\epsilon I \end{bmatrix},$$

when l is odd,

$$\begin{aligned} \tilde{\mathcal{H}}_l(y,\epsilon) &= \begin{bmatrix} 0 & -(C_l(y))^* \\ C_l(y) & 0 \end{bmatrix} && 1 \leq l \leq k, \\ \tilde{\mathcal{H}}_l(y,\epsilon) &= 0 && k+1 \leq l < 2k, \end{aligned}$$

and, when l is even,

$$\tilde{\mathcal{H}}_l(y,\epsilon) = \begin{bmatrix} c_l(y)\epsilon I & (C_l(y))^* \\ C_l(y) & 0 \end{bmatrix} \qquad 1 \le l \le k,$$
$$\tilde{\mathcal{H}}_l(x,\epsilon) = \begin{bmatrix} c_l(y)\epsilon I & 0 \\ 0 & 0 \end{bmatrix} \qquad k+1 \le l \le 2k.$$

Proof. The input polynomial can be rearranged as

$$P(x+yi) = \sum_{j=0}^{k} (-y+ix)^j (-i)^j K_j.$$

Therefore by making the substitutions $x = -y$ and replacing K_j by $(-i)^j K_j$ for $j = 1, \ldots, n$ in Theorem 5 it follows that the set of x for which ϵ is a singular value of $\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}$ is the set of x such that $xi \in \Lambda(\tilde{\mathcal{H}}(y,\epsilon))$. Let the rightmost intersection point of the horizontal line with $\Lambda_\epsilon(P,\gamma)$ be $\hat{x}$. Clearly $\hat{x}i \in \Lambda(\tilde{\mathcal{H}}(y,\epsilon))$; therefore the largest x such that $xi \in \Lambda(\tilde{\mathcal{H}}(y,\epsilon))$ is greater than or equal to $\hat{x}$. If such largest x is strictly greater than $\hat{x}$, the strict inequality

$$\sigma_{\min}\left(\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}\right) < \epsilon$$

holds, which contradicts the fact that the minimum singular value of $\frac{P(x+yi)}{p_\gamma(\sqrt{x^2+y^2})}$ is a continuous function of x and approaches a value that is greater than ϵ as $x \to \infty$. □

2.2 Distance to instability

The distances to the closest unstable continuous systems for the systems (2.1) and (2.2) are the norms of the smallest perturbations moving one or more eigenvalues of the matrix A and the matrix polynomial P onto the imaginary axis, respectively. By substituting the imaginary axis for the boundary of the unstable region $\partial\mathbb{C}_b$ in (1.12) and (1.8), the relationship between the ϵ-pseudospectral abscissa and the distance to instability can be stated as

$$\beta_c(A) \le \epsilon \iff \alpha_\epsilon(A) \le 0$$
$$\beta_c(P,\gamma) \le \epsilon \iff \alpha_\epsilon(P,\gamma) \le 0$$

where β_c denotes the continuous distance to instability. For the Grcar matrix and the upper triangular matrix whose pseudospectra are illustrated in Figure 2.1 and Figure 2.2, the distances to instability are 2.97×10^{-4} and 0.15, respectively.

For continuous systems the equivalent characterizations (1.13) and (1.10) reduce to

$$\beta_c(A) = \inf_{\omega \in \mathbb{R}} \sigma_{\min}(A - \omega i I), \tag{2.18}$$

$$\beta_c(P, \gamma) = \inf_{\omega \in \mathbb{R}} \sigma_{\min} \left(\frac{P(\omega i)}{p_\gamma(|\omega|)} \right). \tag{2.19}$$

In [15] Byers introduced a bisection algorithm for the computation of $\beta_c(A)$. The bisection algorithm, given an estimate $\hat{\beta}$, infers whether or not there is an ω satisfying $\sigma_{\min}(A - \omega i I) = \hat{\beta}$ by the existence of an imaginary eigenvalue of the Hamiltonian matrix $V(0, \hat{\beta})$ defined by (2.6). It either updates an upper bound or a lower bound, accordingly.

Bruinsma and Steinbuch in [10] and Boyd and Balakrishnan in [9] replaced the bisection technique with quadratically convergent algorithms. Here we focus on the algorithm of Boyd and Balakrishnan. Algorithm 2 is the general Boyd-Balakrishnan algorithm for the distance to instability applicable whenever for a given estimate $\hat{\beta}$ we are capable of finding all $\lambda \in \partial \mathbb{C}_b$ such that

$$f(\lambda) = \sigma_{\min}(A - \lambda i I) = \hat{\beta}$$

for the first-order system or

$$h(\lambda) = \sigma_{\min} \left(\frac{P(\lambda i)}{p_\gamma(|\lambda|)} \right) = \hat{\beta}$$

for the higher-order system. Once this set is available, by the continuity of the minimum singular value functions with respect to λ, it is straightforward to determine the intervals in which the inequalites $f(\lambda) < \hat{\beta}$ and $h(\lambda) < \hat{\beta}$ hold. The estimate is refined to the minimum value that the function f or h attains over the midpoints of these intervals.

For the system (2.1) all real ω such that $f(\omega i) = \hat{\beta}$ can be found by extracting the set of imaginary eigenvalues of the Hamiltonian matrix $V(0, \hat{\beta})$. Similarly for the higher-order system (2.1), the set of real ω satisfying $h(\omega i) = \hat{\beta}$ correspond to the imaginary parts of the imaginary eigenvalues of the $*$-even matrix polynomial $\mathcal{V}(0, \hat{\beta})$ (see Theorem 5 for the definition of the matrix polynomial $\mathcal{V}$). In [9] the Boyd-Balakrishnan algorithm is proved to be quadratically convergent for the first-order continuous system.

Algorithm 2 Generic Boyd-Balakrishnan algorithm for the distance to instability

Call:	$\hat{\beta} \leftarrow$ `distinstab`(A,tol) or $\hat{\beta} \leftarrow$ `distinstab`(P,γ,tol) .
Input:	$A \in \mathbb{C}^{n\times n}$ or $P \in \mathbb{C}^{k\times n\times n}$ (matrix polynomial), and $\gamma \in \mathbb{R}_+^{k+1}$ and nonzero (scaling vector), $tol \in \mathbb{R}_+$ (tolerance for termination).
Output:	$\hat{\beta} \in \mathbb{R}_+$, the estimate value for the distance to instability of A or P subject to perturbations determined by γ.

Set $j = 0$, $\Phi^0 = [\lambda_0]$ where $\lambda_0 \in \partial\mathbb{C}_b$ and $\hat{\beta}^0 = \infty$.
repeat

Update the estimate for the distance to instability: Refine the estimate for the distance to instability for the first-order system to

$$\hat{\beta}^{j+1} = \min\{f(\lambda) : \lambda \in \Phi^j\} \tag{2.20}$$

or the estimate for the distance to instability for the higher-order system to

$$\hat{\beta}^{j+1} = \min\{h(\lambda) : \lambda \in \Phi^j\}. \tag{2.21}$$

Update the set of the midpoints: Determine the set of $\lambda \in \partial\mathbb{C}_b$ satisfying $f(\lambda) = \hat{\beta}^{j+1}$ or $h(\lambda) = \hat{\beta}^{j+1}$. From these infer the open intervals $\mathcal{I}_d^{j+1} = (l_d^{j+1}, u_d^{j+1})$, for $d = 1, \ldots, m^{j+1}$ such that $\forall\lambda \in \mathcal{I}_d^{j+1}$, $f(\lambda) < \hat{\beta}^{j+1}$ or $h(\lambda) < \hat{\beta}^{j+1}$. Calculate the new set of midpoints

$$\Phi^{j+1} = \{\frac{u_d^{j+1} + l_d^{j+1}}{2},\ \ d = 1, \ldots, m^{j+1}\}.$$

Increment j.
until $\hat{\beta}_j - \hat{\beta}_{j-1} < tol$.
return $\hat{\beta}_j$

2.3 Numerical examples

In the next subsection we compute the ratio $\alpha_\epsilon(A)/\epsilon$ for various ϵ using Algorithm 1 for the Grcar and the upper triangular matrices discussed in §2.1. Then in §2.3.2 we illustrate that quadratic convergence is achieved for the algorithms in §2.1 and §2.2 in practice. In §2.3.3 the running times of the algorithms are provided for Demmel matrices of various size. Finally the last subsection is devoted to examples for matrix polynomials.

2.3.1 Bounding the continuous Kreiss constant

One of the reasons for our interest in the computation of the ϵ-pseudospectral abscissa is because it appears in the Kreiss constant

$$\mathcal{K}_c(A) = \sup_{\epsilon>0} \frac{\alpha_\epsilon(A)}{\epsilon}, \tag{2.22}$$

which is a good estimator of the magnitude of the maximum transient peak of the continuous system as revealed by (2.5). In Figure 2.4, $\alpha_\epsilon(A)/\epsilon$ is plotted as a function of ϵ for the Grcar matrix on the top and the upper triangular matrix at the bottom. For these plots $\alpha_\epsilon(A)$ is computed for various ϵ using Algorithm 1. In both of the figures, for all ϵ such that $\alpha_\epsilon(A)$ is negative, we replace the ratio by zero for convenience as the negative values are irrelevant for the transient peak. The computed values are also listed in Table 2.1.

We see that for the Grcar matrix example the supremum is achieved around $\epsilon = 10^{-3}$, which is slightly greater than the distance to instability of the matrix. For the upper triangular matrix the supremum is equal to one and attained in the limit as $\epsilon \to \infty$, consistent with the good transient behavior of the upper triangular matrix.

2.3.2 Quadratic convergence

The algorithms in §2.1 and §2.2 are proved to be quadratically convergent in [12] and [9], respectively. For a 10×10 companion matrix example available in *EigTool*'s demo menu, shifted by $-3.475I$, and $\epsilon = 10^{-5}$, the algorithm for the pseudospectral abscissa converges in four iterations. In Table 2.2 the number of accurate digits of the estimate for the pseudospectral abscissa is at least doubled at each iteration until the error in the estimate is close to machine precision. Similarly in Table 2.3 the number of accurate digits of the estimate is at least doubled at the 7th, 8th and 9th iterations when the distance to instability of the companion matrix is computed using the Boyd-Balakrishnan algorithm. This

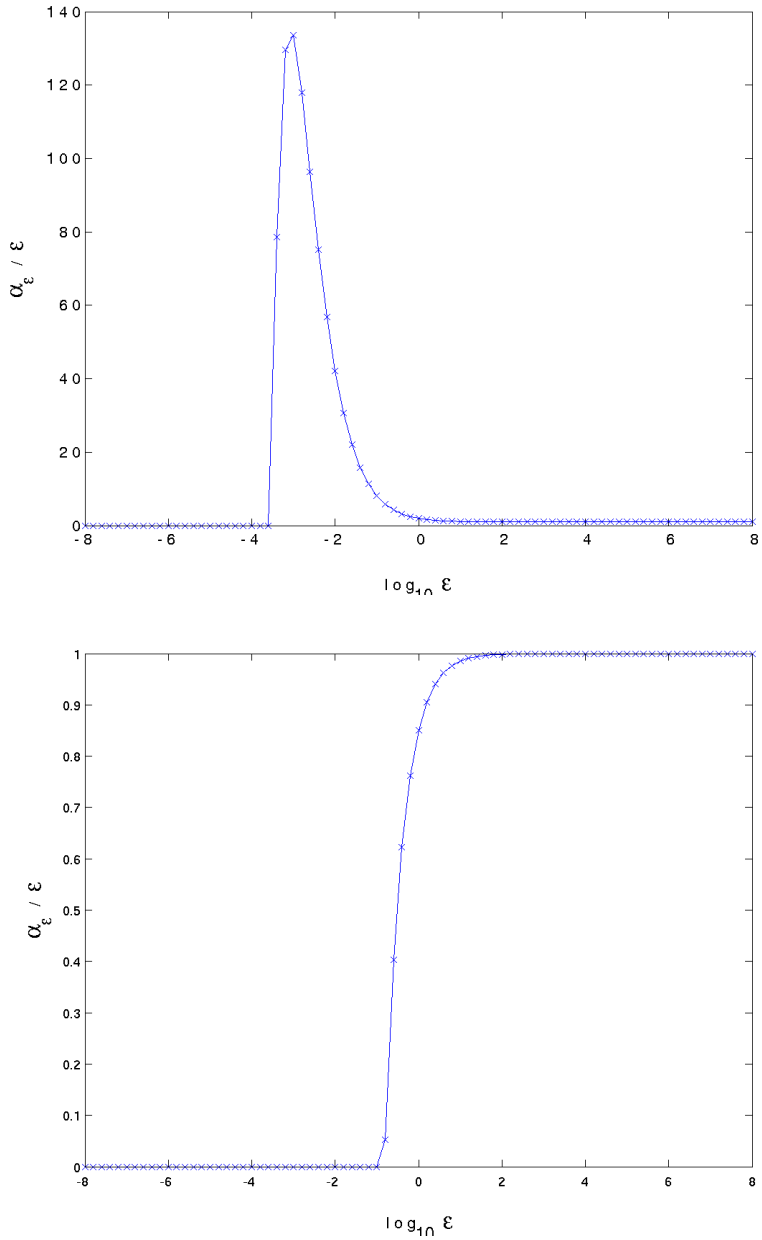

Figure 2.4: The ratio α_ϵ/ϵ is plotted as a function of $log_{10}\epsilon$ for the Grcar matrix on the top and the upper triangular matrix at the bottom.

ϵ	$\frac{\alpha_\epsilon}{\epsilon}$ for the Grcar matrix	$\frac{\alpha_\epsilon}{\epsilon}$ for the upper triangular matrix
10^{-4}	$-1.125076668581613e+003$	$-1.575128249363217e+003$
10^{-3}	$1.336232734017432e+002$	$-1.526302151021469e+002$
10^{-2}	$4.206404810678649e+001$	$-1.408713338112931e+001$
10^{-1}	$8.070545282717980e+000$	$-5.010790044998323e-001$
1	$1.913868744168375e+000$	$8.499889226137701e-001$
10	$1.096897359709284e+000$	$9.849998889272065e-001$
10^2	$1.009758733899733e+000$	$9.984999988889766e-001$
10^3	$1.000976583115880e+000$	$9.998499999888924e-001$
10^4	$1.000097665429625e+000$	$9.999849999998980e-001$
10^5	$1.000009766614169e+000$	$9.999985000000028e-001$
10^6	$1.000000976662132e+000$	$9.999998500000047e-001$
10^7	$1.000000976662132e+000$	$9.999999850000044e-001$

Table 2.1: The ratios α_ϵ/ϵ for the Grcar matrix and the upper triangular matrix.

Iteration	Estimate x^j for α_ϵ
0	-0.100129771527858
1	1.066949720709850
2	1.085214307719376
3	1.085216433113323
4	1.085216433113349

Table 2.2: The values of the estimate for the ϵ-pseudospectral abscissa at each iteration for a companion matrix example and $\epsilon = 10^{-5}$ are given. The algorithm reaches the limits of the machine precision in four iterations.

example is an extreme one for the computation of the distance to instability; only after the first six iterations do we observe rapid convergence.

2.3.3 Running times

We ran the algorithms for the ϵ-pseudospectral abscissa and the continuous distance to instability on Demmel matrices [21] of various size and $\epsilon = 10^{-2}$. Demmel matrices are upper triangular Toeplitz matrices. For all j the ratio $a_{1,j+1}/a_{1,j}$ is a constant greater than one and chosen so that $a_{1,1} = -1$ and $a_{1,n} = -10^{-4}$. From Tables 2.4 and 2.5 it is apparent that the running time is cubic with respect to the size of the input matrix, as the computations are

Iteration	Estimate $\hat{\beta}^j$ for β_c
0	$7.638415697927511e-04$
1	$1.914153978309624e-04$
2	$5.526298624300591e-05$
3	$1.968319726613891e-05$
4	$8.451406155777241e-06$
5	$3.015678091250249e-06$
6	$8.555365518847272e-07$
7	$7.499990333283894e-07$
8	$7.499529185454231e-07$
9	$7.499529185323792e-07$

Table 2.3: The values of the estimate for the continuous distance to instability at each iteration for the companion matrix example are given. The algorithm reaches the limits of the machine precision in nine iterations.

Size	Total running time in secs	Running time per iteration in secs.
10	0.070	0.012
20	0.180	0.026
40	1.030	0.129
80	5.990	0.856
160	54.960	7.851
320	574.500	71.8125

Table 2.4: The total and average running times in seconds per iteration of Algorithm 1 on the Demmel example of various size and $\epsilon = 10^{-2}$.

dominated by the solution of the Hamiltonian eigenvalue problems of double size. In the tables both the overall running times and the average running times per iteration are listed. In general the computation of the distance to instability requires less time, since for the pseudospectral abscissa we need to apply both the vertical and the horizontal searches, while for the distance to instability only the level sets of the minimum singular value function need to be determined by solving Hamiltonian eigenvalue problems.

2.3.4 Matrix polynomials

The algorithm in §2.1.3 for matrix polynomials converges to the ϵ-pseudospectral abscissa rapidly but with a cost of $O(n^3k^3)$ (the time required to solve $*$-even

Size	Total running time in secs	Running time per iteration in secs.
10	0.070	0.007
20	0.170	0.015
40	0.650	0.059
80	3.290	0.329
160	34.520	3.452
320	374.450	34.041

Table 2.5: The total and average running time per iteration of the Boyd-Balakrishnan algorithm in §2.2 for the continuous distance to instability on the Demmel example of various size.

polynomial eigenvalue problems of size $2n$ and degree $2k$) at each iteration. Consider the quadratic 4×4 matrix polynomial $Q(\lambda) = \sum_{j=0}^{2} \lambda^j Q_j$ with

$$Q_2 = \begin{bmatrix} -12 & -36 & -72 & -72 \\ 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 03 & 0 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \text{ and}$$
$$Q_0 = \begin{bmatrix} -3-i & -0.5i & -1/3i & -2.5i \\ \pi & -3-i & -0.5i & -1/3i \\ i & \pi & -3-i & -0.5i \\ 0.5i & i & \pi & -3-i \end{bmatrix}. \tag{2.23}$$

The ϵ-pseudospectra of Q with the scaling vector $\gamma = [1\ 1\ 1]$ for $\epsilon = 0.1$, 0.3, 0.5, 0.7, 0.9 and the scaling vector $\gamma = [0.1\ 1\ 0.1]$ for $\epsilon = 1$, 3, 5, 7, 9 are displayed in Figure 2.5 on the top and at the bottom, respectively. The ϵ-psedospectral abscissa values are computed by Algorithm §2.1.3 and the points where the ϵ-pseudospectral abscissas are attained are marked by black circles in the figures. In Table 2.6 the estimates generated by the algorithm are given for $\gamma = [0.1\ 1\ 0.1]$ and $\epsilon = 3$. We again observe fast convergence; in particular at the 2nd and 3rd iterations the precision of the estimate is doubled.

Quadratic matrix polynomials with positive definite Hermitian coefficients are stable as all of the eigenvalues are contained in the left half-plane [67]. For the quadratic matrix polynomial $\tilde{Q}(\lambda) = \sum_{j=0}^{2} \lambda \tilde{Q}_j$ with the positive definite

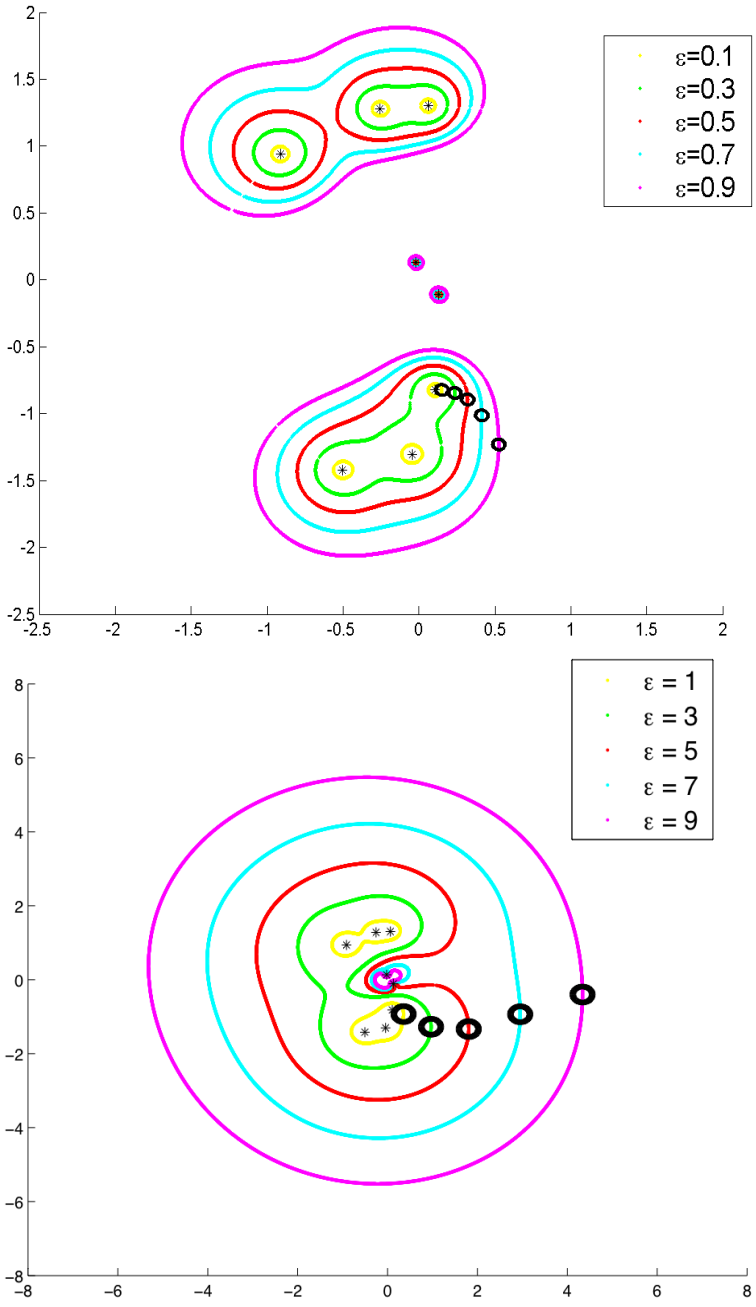

Figure 2.5: The ϵ-pseudospectrum of Q defined by (2.23) is displayed for $\gamma = [1\ 1\ 1]$ and $\epsilon = 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9$ on the top and for $\gamma = [0.1\ 1\ 0.1]$ and $\epsilon = 1,\ 3,\ 5,\ 7,\ 9$ at the bottom. The black circles and asterisks mark the points furthest to the right on each ϵ-pseudospectrum and the eigenvalues.

Iteration	Estimate x^j for $\alpha_\epsilon(Q,\gamma)$
0	0.131033609557413
1	0.964129453835321
2	0.969445951451222
3	0.969446006137978
4	0.969446006137979

Table 2.6: The value of the estimate for the ϵ-pseudospectral abscissa at each iteration are given for the matrix polynomial Q, $\gamma = [0.1\ 1\ 0.1]$ and $\epsilon = 3$.

Hermitian coefficients

$$\tilde{Q}_2 = \begin{bmatrix} 124 & 33 & 72 & 72 \\ 33 & 100 & -3 & 0 \\ 72 & -3 & 100 & -3 \\ 72 & 0 & -3 & 100 \end{bmatrix}, \quad \tilde{Q}_1 = \begin{bmatrix} 7.2 & -6 & -2 & -1 \\ -6 & 9.2 & -4 & -1 \\ -2 & -4 & 11.2 & -2 \\ -1 & -1 & -2 & 13.2 \end{bmatrix} \text{ and}$$
$$\tilde{Q}_0 = \begin{bmatrix} 9 & -\pi + 0.5i & 4/3i & 0.75i \\ -\pi - 0.5i & 9 & -\pi + 0.5i & 4/3i \\ -4/3i & -\pi - 0.5i & 9 & -\pi + 0.5i \\ -0.75 & -4/3i & -\pi - 0.5i & 9 \end{bmatrix} \tag{2.24}$$

plots of the function $h(\omega) = \sigma_{\min}(P(\omega)/p_\gamma(|\omega|)),\ \omega \in \mathbb{R}$ are provided for $\gamma = [0.1\ 1\ 1]$, $\gamma = [0.3\ 1\ 1]$ and $\gamma = [0.7\ 1\ 1]$ in Figure 2.6. The figure on the top displays the functions in the interval $[-0.5, 0.5]$, while the figure at the bottom displays the same functions over the interval $[-5, 5]$. In the figure on the top we use black asterisks to indicate the points where the functions are minimized (equivalently, where the distances to instability are attained). It is apparent from the figures that the function $h(\omega)$ for small ω in absolute value is very sensitive to changes in the scaling α_0, while the same function is very insensitive to changes in the scalings to the other coefficients. This is justified by the fact that the eigenvalues closest to the imaginary axis that are of interest have small moduli. These eigenvalues are more sensitive to perturbations to Q_0 rather than perturbations to $\tilde{Q}_1$ and $\tilde{Q}_2$. Furthermore, the eigenvalues of the matrix $\tilde{Q}_0$ are highly ill-conditioned. For large ω in absolute value the functions $h(\omega)$ with the scalings $\gamma = [0.1\ 1\ 1]$, $\gamma = [0.3\ 1\ 1]$ and $\gamma = [0.7\ 1\ 1]$ at the bottom in Figure 2.6 are similar and in the limit as $\omega \to \infty$ and $\omega \to -\infty$ they become identical. Notice also that the function $h(\omega)$ is nonconvex and nonsmooth around the origin, so standard smooth optimization techniques may fail to locate the global minimum. The fast convergence of Algorithm 2 to compute $\beta_c(\tilde{Q}, \gamma)$ on this example, with $\gamma = [0.3\ 1\ 1]$, is illustrated in Table 2.7.

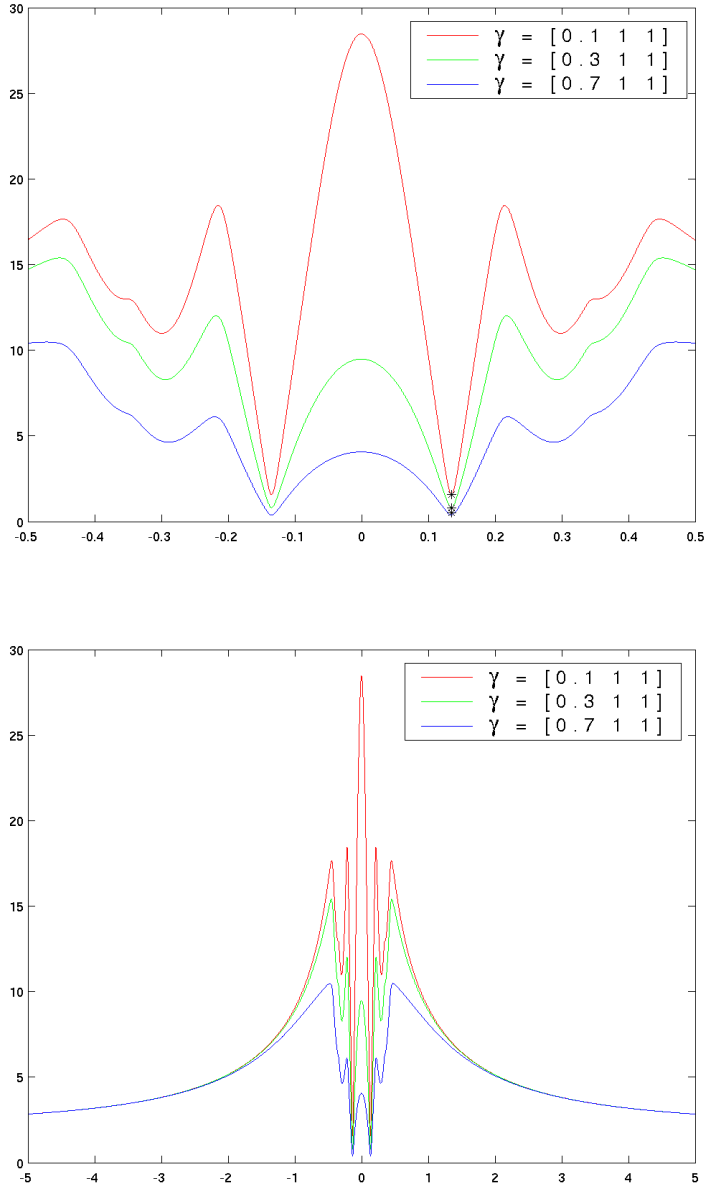

Figure 2.6: The function $h(\omega)$ for the quadratic matrix polynomial $\tilde{Q}$ with $\gamma = [0.1\ 1\ 1]$, $\gamma = [0.3\ 1\ 1]$ and $\gamma = [0.7\ 1\ 1]$ are displayed. On the top and at the bottom the functions are shown over the intervals $[-0.5, 0.5]$ and $[-5, 5]$, respectively. The black asterisks show the global minimizers.

Iteration	Estimate x^j for $\beta_c(\tilde{Q}, \gamma)$
0	$2.067475497667851e+000$
1	$8.174502767041282e-001$
2	$8.127462094636189e-001$
3	$8.127461887310047e-001$

Table 2.7: The values of the estimates generated by the Boyd-Balakrishnan algorithm for the continuous distance to instability of $\tilde{Q}$ with $\gamma = [0.3\ 1\ 1]$ indicate rapid convergence.

Chapter 3

Robust Stability Measures for Discrete Systems

In this chapter we present algorithms for the computation of various measures for the robust stability and the initial behavior of the autonomous first-order discrete-time dynamical system

$$x_{k+1} = Ax_k \tag{3.1}$$

and the higher-order discrete-time dynamical system

$$K_k x_{j+k} + K_{k-1} x_{j+k-1} + \cdots + K_0 x_j = 0. \tag{3.2}$$

The algorithms that we introduce in §3.1 for the pseudospectral radius of a matrix and a matrix polynomial are the first efficient techniques for high-precision computation. The numerical radius algorithm for a matrix in §3.2 combines the ideas in [9] and [34]. Finally in §3.3 we briefly specify the details of the Boyd-Balakrishnan algorithm for the distance to instability for discrete first-order and higher-order systems. The algorithms for the pseudospectral radius and numerical radius have also been described in [59].

3.1 Pseudospectral radius

We start with the first-order system and extend the ideas to the higher-order system in the last subsection. For the first-order discrete system the point in the ϵ-pseudospectrum furthest away from the origin, called the *ϵ-pseudospectral radius*,

$$\rho_\epsilon(A) = \max\{|z| : z \in \Lambda_\epsilon(A)\}, \tag{3.3}$$

is useful for estimating the norms of the powers of A. Figure 3.1 illustrates the pseudospectra of a random matrix (*i.e.* the real and imaginary components of

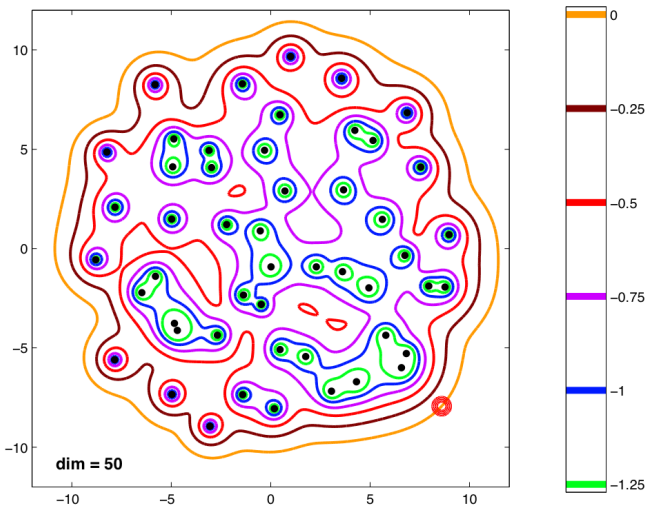

Figure 3.1: The eigenvalues (solid dots) and the ϵ-pseudospectra of a random $50\times$ 50 matrix are shown for various values of ϵ. The bar on the right shows the values of ϵ on a log 10 scale. The point where the ϵ-pseudospectral radius is attained for $\epsilon = 1$ is in the lower right corner and is marked with a circle.

the entries of the matrix are chosen from a normal distribution with mean 0 and standard deviation 1 independently) for various values of ϵ together with the point in the ϵ-pseudospectrum with the largest modulus for $\epsilon = 1$.

For the pseudospectral radius it can be deduced from the Kreiss matrix theorem [44, 70, 72] that

$$\sup_{\epsilon>0} \frac{\rho_\epsilon(A) - 1}{\epsilon} \leq \sup_k \|A^k\| \leq en \sup_{\epsilon>0} \frac{\rho_\epsilon(A) - 1}{\epsilon}. \tag{3.4}$$

In (3.4) the supremum of the norms of the matrix powers is bounded above and below in terms of the ϵ-pseudospectral radius. The lower bound is especially useful as an indicator of how large the norms of the matrix powers grow. In Figure 2.2 we have seen that the real parts of the eigenvalues of the upper triangular matrix with the entries $a_{ij} = -0.3,\ j \geq i$ are not sensitive to perturbations. In the same figure we observe that the moduli of the eigenvalues increase rapidly as the norms of the perturbations increase. Perturbations with norms 10^{-7} move the eigenvalues outside of the unit circle. More precisely, for $\epsilon = 10^{-7}$ the ϵ-pseudospectral radius is 1.06 and the lower bound in (3.4) indicates that the norm of the matrix powers must exceed 6×10^5.

In this section we put emphasis on the computation of the pseudospectral radius. We will exploit the singular value characterization (1.17) and also refer

to the strict ϵ-pseudospectrum

$$\Lambda'_\epsilon(A) = \{z \in \mathbb{C} : \sigma_{\min}(A - zI) < \epsilon\}. \tag{3.5}$$

For the convergence of continuous-time systems the analogous quantity to the pseudospectral radius is the pseudospectral abscissa, the maximum of the real parts of the points in the pseudospectrum. We described in the previous chapter the quadratically convergent algorithm to compute the pseudospectral abscissa from [11]. A first thought to compute the pseudospectral radius of A might be to reduce the problem to the computation of the pseudospectral abscissa of a related matrix. Given a complex number $re^{i\theta}$ in the pseudospectrum, by taking the logarithm, we obtain $\ln r + i\theta$. Denoting the set that is obtained by taking the logarithm of every point in $\Lambda_\epsilon(A)$ by $\ln(\Lambda_\epsilon(A))$, we conclude that the real part of the rightmost point in $\ln(\Lambda_\epsilon(A))$ is equal to the logarithm of the pseudospectral radius of A. However, there may not be any matrix with the pseudospectrum $\ln(\Lambda_\epsilon(A))$. For instance, as we discuss later in this section, there are matrices for which the boundary of the ϵ-pseudospectrum contains an arc of a circle centered at the origin. For such a matrix A, a line parallel to the imaginary axis intersects the boundary of the set $\ln(\Lambda_\epsilon(A))$ at infinitely many points, but in [11], it is shown that vertical cross sections of the ϵ-pseudospectrum of a matrix have only finitely many boundary points. Therefore, we derive an algorithm tailored to the pseudospectral radius, following the ideas in [11].

Before presenting a locally quadratically convergent algorithm for the ϵ-pseudospectral radius in §3.1.2 and §3.1.3, we discuss its variational properties. The convergence analysis of the algorithm is similar to that in [12], so we briefly justify our claims about its convergence properties in §3.1.4. The boundary of the pseudospectrum of a matrix may contain arcs of circles which may potentially cause numerical trouble for the pseudospectral radius algorithm. We investigate this phenomenon in §3.1.5. In §3.1.6 we specify a version of the algorithm in floating point arithmetic which is expected to produce accurate results as long as the pseudospectral radius problem is well conditioned. Finally, the extension of the algorithm to higher-order systems is discussed in §3.1.7.

3.1.1 Variational properties of the ϵ-pseudospectral radius

In this subsection we are interested in how the pseudospectral radius $\rho_\epsilon(X)$ varies with respect to ϵ and X. Thus we view the pseudospectral radius as a mapping from $\mathbb{R}_+ \times \mathbb{C}^{n\times n}$ to $\mathbb{R}_+$. The most basic result we are looking for is the continuity of the pseudospectral radius with respect to ϵ and X. For this purpose, notice that the pseudospectral radius is the robust regularization of

the spectral radius in the sense of [51], *i.e.*

$$\rho_\epsilon(X) = \sup_Y \{\rho(Y) \ : \ \|Y - X\| \leq \epsilon\}. \tag{3.6}$$

Since the spectral radius is continuous in matrix space, the continuity of $\rho_\epsilon(X)$ in matrix space immediately follows from Proposition 3.5 in [51]. In fact, joint continuity with respect to X and ϵ can also be shown by proving upper and lower semicontinuity separately [50] :

Theorem 7 (**A.S. Lewis**). *The function $\rho_\epsilon(X)$ is jointly continuous with respect to ϵ and X everywhere.*

The next result states that the $(\epsilon + \beta)$-pseudospectral radius of X depends on the ϵ-pseudospectral radius of the matrices in the β neighborhood of X.

Theorem 8. *Let β and ϵ be nonnegative real numbers. Then*

$$\rho_{\epsilon+\beta}(X) = \sup_{\|X'-X\| \leq \beta} \rho_\epsilon(X').$$

Proof. By definition (3.6)

$$\begin{aligned} \rho_{\epsilon+\beta}(X) &= \sup_Y \{\rho(Y) : \|Y - X\| \leq \epsilon + \beta\} \\ &= \sup_{Y,X'} \{\rho(Y) : \|X' - X\| \leq \beta, \|Y - X'\| \leq \epsilon\} \\ &= \sup_{X'} \{\rho_\epsilon(X') : \|X' - X\| \leq \beta\}. \end{aligned}$$

Therefore the result follows. □

Next we focus on the differentiability of $\rho_\epsilon(X)$. For this purpose let us introduce the function $p_{(\epsilon,X)} : \ \mathbb{R}_+ \times [0, 2\pi) \to \mathbb{R}$ for a given ϵ and X defined by

$$p_{(\epsilon,X)}(r, \theta) = \sigma_{\min}(X - re^{i\theta} I) - \epsilon. \tag{3.7}$$

Note that $p_{(\epsilon,X)}(r, \theta)$ is less than or equal to 0 if and only if the complex number $re^{i\theta}$ belongs to the ϵ-pseudospectrum of X. Well known properties of the minimum singular value function imply that $p_{(\epsilon,X)}(r, \theta)$ is a continuous function of r and θ. The theorem below specifies the conditions under which the function $p_{(\epsilon,X)}(r, \theta)$ is differentiable with respect to r and θ. A real-valued function defined on a real domain is called real-analytic at a given point if the function has a real convergent Taylor expansion at the given point.

Theorem 9. *Let $\epsilon \in \mathbb{R}_+$ and $X \in \mathbb{C}^{n\times n}$. If the minimum singular value of $X - re^{i\theta}I$ is greater than 0 and has multiplicity one, then at (r,θ) the function $p_{(\epsilon,X)}(r',\theta')$ is real-analytic with respect to r' and θ' with derivatives*

$$\nabla p_{(\epsilon,X)}(r,\theta) = (-\text{Re } e^{i\theta}u^*v, \text{Im } re^{i\theta}u^*v)$$

where u and v are any consistent pair of unit left and right singular vectors corresponding to $\sigma_{\min}(X - re^{i\theta}I)$.

Proof. The function $\sigma_{\min}(X - r'e^{i\theta'}I)$ is real-analytic at (r,θ) provided $\sigma_{\min}(X - re^{i\theta}I)$ is positive and has multiplicity one. This immediately follows from the fact that $X_2(r',\theta') = (X^* - r'e^{-i\theta'}I)(X - r'e^{i\theta'}I)$ is analytic with respect to r' and θ' and therefore $\sigma^2_{\min}(X - r'e^{i\theta'}I)$, the smallest eigenvalue of $X_2(r',\theta')$, is real analytic whenever $\sigma^2_{\min}(X - r'e^{i\theta'}I)$ has multiplicity one. The derivatives can be derived by applying the chain rule to the result of Theorem 7.1 in [11]. □

For a fixed ϵ and X, we call the constrained optimization problem

$$\sup_{p_{(\epsilon,X)}(r,\theta)\le 0} r \tag{3.8}$$

the *ϵ-pseudospectral radius problem at X*. By the definition of $p_{(\epsilon,X)}(r,\theta)$ and the definition of the ϵ-pseudospectral radius given in (1.17), we see that the value attained at a global maximizer of the ϵ-pseudospectral radius problem at X is equal to $\rho_\epsilon(X)$. Now we are ready to derive the derivatives of $\rho_\epsilon(X)$ with respect to ϵ and X

Theorem 10. *Let a matrix $X_0 \in \mathbb{C}^{n\times n}$ and $\epsilon_0 \in \mathbb{R}_+$ be given. Suppose that (r_0,θ_0) is a local maximizer of the ϵ_0-pseudospectral radius problem at X_0 and the multiplicity of $\sigma_{\min}(X_0 - r_0e^{i\theta_0}I)$ is one. Then the gradient of $p_{(\epsilon_0,X_0)}(r,\theta)$ at (r_0,θ_0) is a nonnegative multiple of $(1,0)$.*

Moreover, if the point (r_0,θ_0) is the unique global maximizer, $\nabla p_{(\epsilon_0,X_0)}(r_0,\theta_0)$ is nonzero and the Hessian of p with respect to r and θ, $\nabla^2 p_{(\epsilon_0,X_0)}(r_0,\theta_0)$, is nonsingular, then at (ϵ_0,X_0) the function $\rho_\epsilon(X)$ is differentiable with respect to ϵ and X with derivatives

$$\frac{d\rho_{\epsilon_0}(X_0)}{d\epsilon} = \frac{-1}{\text{Re } e^{i\theta_0}u^*v} \quad , \quad \nabla_X \rho_{\epsilon_0}(X_0) = \frac{uv^*}{\text{Re } e^{i\theta_0}u^*v},$$

where u and v are any consistent pair of unit left and right singular vectors corresponding to $\sigma_{\min}(X_0 - r_0e^{i\theta_0}I)$.

Proof. By assumption (r_0,θ_0) is a local maximizer of the ϵ_0-pseudospectral radius problem at X_0 and by Theorem 9, $p_{(\epsilon_0,X_0)}(r,\theta)$ is differentiable with respect

to r and θ at this maximizer. Therefore, provided the gradient of $p_{(\epsilon_0,X_0)}(r,\theta)$ is nonzero, standard first-order necessary conditions must be satisfied. Thus either the gradient of $p_{(\epsilon_0,X_0)}$ at (r_0,θ_0) is 0 or there exists a positive μ such that

$$(1,0) - \mu\nabla p_{(\epsilon_0,X_0)}(r_0,\theta_0) = 0.$$

In either case the gradient is a nonnegative multiple of $(1,0)$ as desired. From Theorem 9, we know that

$$\nabla p_{(\epsilon_0,X_0)}(r_0,\theta_0) = (-\text{Re } e^{i\theta_0}u^*v, \text{Im } r_0e^{i\theta_0}u^*v),$$

so when $u^*v \neq 0$, we have $\mu = \frac{-1}{\text{Re } e^{i\theta_0}u^*v}$.

When (r_0,θ_0) is the unique global maximizer with nonzero gradient and nonsingular Hessian, we deduce from a standard sensitivity result such as Theorem 5.53 in [8] that $\frac{d\rho_{\epsilon_0}(X_0)}{d\epsilon} = -\mu\frac{dp_{(\epsilon_0,X_0)}(r_0,\theta_0)}{d\epsilon} = \mu$ and $\nabla_X\rho_{\epsilon_0}(X_0) = -\mu\nabla_X p_{(\epsilon_0,X_0)}(r_0,\theta_0) = -\mu uv^*$ hold (since $\nabla_X p_{(\epsilon_0,X_0)}(r_0,\theta_0) = uv^*$; see Theorem 7.1 in [11]). □

3.1.2 Radial and circular searches

The algorithm depends on the steps that we call circular and radial searches. Figure 3.2 illustrates a radial and a circular search for a variant of a 3×3 example given by Demmel [21] and for $\epsilon = 10^{-3.18}$. This matrix is an upper triangular Toeplitz matrix with the entry $d_{j,k}$, $k > j$, equal to $-10^{2(k-j)}$ and the entries on the diagonal equal to $0.1 + 0.01i$. For the rest of this section, let us fix $\epsilon \in \mathbb{R}_+$ and the matrix $A \in \mathbb{C}^{n\times n}$ for which we are computing the pseudospectral radius. We drop the subscripts of the function $p_{(\epsilon,A)}(r,\theta)$ for convenience.

The aim of a *radial search* is to find the point on the boundary of the ϵ-pseudospectrum with the largest modulus in a given direction. More formally, given $\theta \in [0,2\pi)$ such that there exists a positive real number r' satisfying $p(r',\theta) = 0$, we want to calculate

$$\eta_\epsilon(\theta) = \max\{r \in \mathbb{R}_+ : p(r,\theta) = 0\}. \tag{3.9}$$

We will state a theorem which suggests how we can compute the r values such that $p(r,\theta) = 0$ holds for a fixed $\theta \in [0,2\pi)$.

Theorem 11. *Let $r, \epsilon \in \mathbb{R}_+$ and $\theta \in [0,2\pi)$. The matrix $A - re^{i\theta}I$ has ϵ as one of its singular values if and only if the matrix*

$$K(\theta,\epsilon) = \begin{bmatrix} ie^{i\theta}A^* & \epsilon I \\ -\epsilon I & ie^{-i\theta}A \end{bmatrix} \tag{3.10}$$

has the pure imaginary eigenvalue ir.

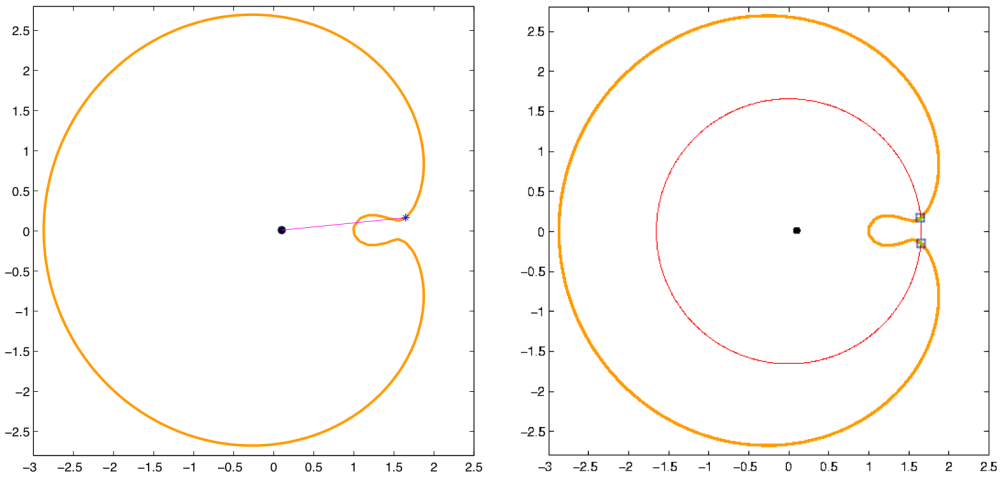

Figure 3.2: The boundary of the ϵ-pseudospectrum for an example due to Demmel. a) The radial search finds the point with the maximum modulus on the pseudospectrum boundary in a given search direction. b) The circular search determines the intersection points of the ϵ-pseudospectrum boundary with a circle of given radius.

Proof. The matrices $A - re^{i\theta}I$ and $iAe^{-i\theta} - irI$ have the same set of singular values. It follows from [15](Theorem 1) and [11](Lemma 5.3) that the matrix $iAe^{-i\theta} - irI$ has the singular value ϵ if and only if the imaginary number ir is an eigenvalue of the matrix in (3.10). □

We note that the matrix $K(\theta, \epsilon)$ is Hamiltonian, *i.e.* $JK(\theta, \epsilon)$ is Hermitian where J is defined by (A.1). By definition (3.9) and Theorem 11, $\eta_\epsilon(\theta)i$ is an imaginary eigenvalue of $K(\theta, \epsilon)$. According to the next corollary $\eta_\epsilon(\theta)i$ is actually the imaginary eigenvalue with the largest imaginary part.

Corollary 12 (Radial Search). *Given a number $\theta \in [0, 2\pi)$ with $p(r', \theta) = 0$ for some r', the quantity $\eta_\epsilon(\theta)$ defined in (3.9) is the largest of the imaginary parts of the pure imaginary eigenvalues of $K(\theta, \epsilon)$.*

Proof. Since there exists r' such that $p(r', \theta) = 0$, Theorem 11 implies that the matrix $K(\theta, \epsilon)$ has an imaginary eigenvalue. Let $r_\epsilon(\theta)i$ be the imaginary eigenvalue of the matrix $K(\theta, \epsilon)$ with greatest imaginary part. By definition (3.9) and Theorem 11, $\eta_\epsilon(\theta)i \in \Lambda(K(\theta, \epsilon))$, *i.e.* $r_\epsilon(\theta) \geq \eta_\epsilon(\theta)$. Now suppose that $r_\epsilon(\theta)$ is strictly greater than $\eta_\epsilon(\theta)$. Again from Theorem 11, we deduce that $A - r_\epsilon(\theta)e^{i\theta}I$ has a singular value ϵ (not necessarily the smallest one), so $p(r_\epsilon(\theta), \theta) \leq 0$. Since p is a continuous function of r and $p(r, \theta)$ approaches ∞

as r goes to ∞, from the intermediate value theorem we conclude that for some $\hat{r} \geq r_\epsilon(\theta) > \eta_\epsilon(\theta)$, $p(\hat{r}, \theta) = 0$. But this contradicts the definition of $\eta_\epsilon(\theta)$ in (3.9). Therefore $\eta_\epsilon(\theta) = r_\epsilon(\theta)$ must hold. □

In a *circular search* we identify the set of points on the boundary of the pseudospectrum with a given modulus. In other words, given a positive real number r, we need to determine those θ values in the interval $[0, 2\pi)$ for which $p(r, \theta) = 0$ is satisfied. A result of Byers [15] implies that $A - e^{i\theta} I$ has ϵ as one of its singular values if and only if the pencil $P(1, \epsilon) - \lambda Q(1, \epsilon)$ has the generalized eigenvalue $e^{i\theta}$ where

$$P(r, \epsilon) = \begin{bmatrix} -\epsilon I & A \\ rI & 0 \end{bmatrix}, \quad Q(r, \epsilon) = \begin{bmatrix} 0 & rI \\ A^* & -\epsilon I \end{bmatrix} \tag{3.11}$$

The pencil $P(r, \epsilon)^* - \lambda Q(r, \epsilon)^*$ is symplectic, *i.e.* $P(r, \epsilon)^* J P(r, \epsilon) = Q(r, \epsilon)^* J Q(r, \epsilon)$ for the matrix J defined in (A.1). Apart from the $*$-symplectic structure of the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$, we note that $D(\theta)(P(r, \epsilon) - e^{i\theta} Q(r, \epsilon))$ is Hermitian for all θ where

$$D(\theta) = \begin{bmatrix} I & 0 \\ 0 & -e^{-i\theta} I \end{bmatrix}. \tag{3.12}$$

The error analysis in §3.1.6 exploits this structure.

We present a generalized version of Byers' result, establishing a relation between the singular values of $A - re^{i\theta} I$ and the eigenvalues of the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$. We recall that a $2n \times 2n$ pencil $X - \lambda Y$ is said to be singular if $\det(X - \lambda Y) = 0$ for all $\lambda \in \mathbb{C}$; otherwise it is said to be regular in which case it has at most $2n$ finite eigenvalues.

Theorem 13 (**Circular Search**). *The matrix $A - re^{i\theta} I$ has ϵ as one of its singular values if and only if the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$ has the generalized eigenvalue $e^{i\theta}$ or the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$ is singular.*

Proof. The matrix $A - re^{i\theta} I$ has the singular value ϵ if and only if

$$\begin{bmatrix} 0 & A - re^{i\theta} I \\ A^* - re^{-i\theta} I & 0 \end{bmatrix}$$

has ϵ as one of its eigenvalues. But this holds if and only if

$$\det \begin{bmatrix} -\epsilon I & A - re^{i\theta} I \\ A^* - re^{-i\theta} I & -\epsilon I \end{bmatrix} = 0$$

or equivalently, multiplying the matrix above by $D^*(\theta)$ on the left,

$$\det \begin{bmatrix} -\epsilon I & A - re^{i\theta} I \\ -e^{i\theta} A^* + rI & \epsilon e^{i\theta} I \end{bmatrix} = 0.$$

By rearranging the matrix above, we see that $\det(P(r, \epsilon) - e^{i\theta} Q(r, \epsilon)) = 0$. □

Unlike in a radial search, in a circular search we wish to determine all of the zeros of $p(r,\cdot)$. Hence, as long as $P(r,\epsilon) - \lambda Q(r,\epsilon)$ is regular, to find the points on the ϵ-pseudospectrum boundary with modulus r we need to check whether $A - re^{i\theta}I$ has ϵ as its minimum singular value for each $\theta \in [0, 2\pi)$ such that $e^{i\theta} \in \Lambda(P(r,\epsilon), Q(r,\epsilon))$.

3.1.3 The algorithm

We now combine radial and circular searches to obtain an algorithm for the ϵ-pseudospectral radius. For now, we assume that the pencil we use for circular searches is regular for all values of r. The issue of singular pencils is the theme of §3.1.5. In particular, we explain how the algorithm below can be modified for singular pencils.

Algorithm 3 is based on the Boyd-Balakrishnan algorithm [9] and the criss-cross method for the pseudospectral abscissa introduced by Burke *et. al.* [12]. It keeps an estimate of the pseudospectral radius and a set of open "intervals", $I_1^j, I_2^j, \cdots, I_{m_j}^j$. Actually all these are intervals $(\iota_k^j, \zeta_k^j) \subset [0, 2\pi)$ with the possible exception of $I_{m_j}^j$ which may "wrap around the circle", *i.e.*, $I_{m_j}^j = (\iota_{m_j}^j, 2\pi) \cup [0, \zeta_{m_j}^j)$ with $\iota_{m_j}^j > \zeta_{m_j}^j$. Let the real number η^j be the estimate of the pseudospectral radius at the jth iteration and let $\theta \in [0, 2\pi)$. Then for $j > 1$ the point $\eta^j e^{i\theta}$ lies inside the strict pseudospectrum if and only if the angle θ is contained in one of $I_1^j, I_2^j, \cdots, I_{m_j}^j$. In the description of the algorithm we use the notation $x \bmod 2\pi$ which refers to the real number in the interval $[0, 2\pi)$ such that $x = l\ 2\pi + x \bmod 2\pi$ for some integer l.

At each iteration, the algorithm applies a radial search in the direction of the midpoint of each interval. The estimate of the pseudospectral radius is refined to the maximum of the modulus values returned by the radial searches. The open intervals are updated by the application of a circular search. New open intervals contain the angles of the points lying inside the strict pseudospectrum and on the circle with radius equal to the new estimate of the pseudospectral radius. Initially, we start with a radial search in the direction of the angle of an arbitrary eigenvalue whose modulus is equal to the spectral radius.

In Figure 3.3, the first two iterations of a sample run of the algorithm are shown. The initial radial search is followed by a circular search which detects four intersection points. Next we perform two radial searches in the directions of the midpoints of two intervals in which $p(\eta^1, \cdot)$ is negative. The maximum of the values returned by the radial searches is our next estimate η^2 for the ϵ-pseudospectral radius. For the specific example, the input matrix is real, so the values returned by the radial searches are equal. We continue with a circular search as before.

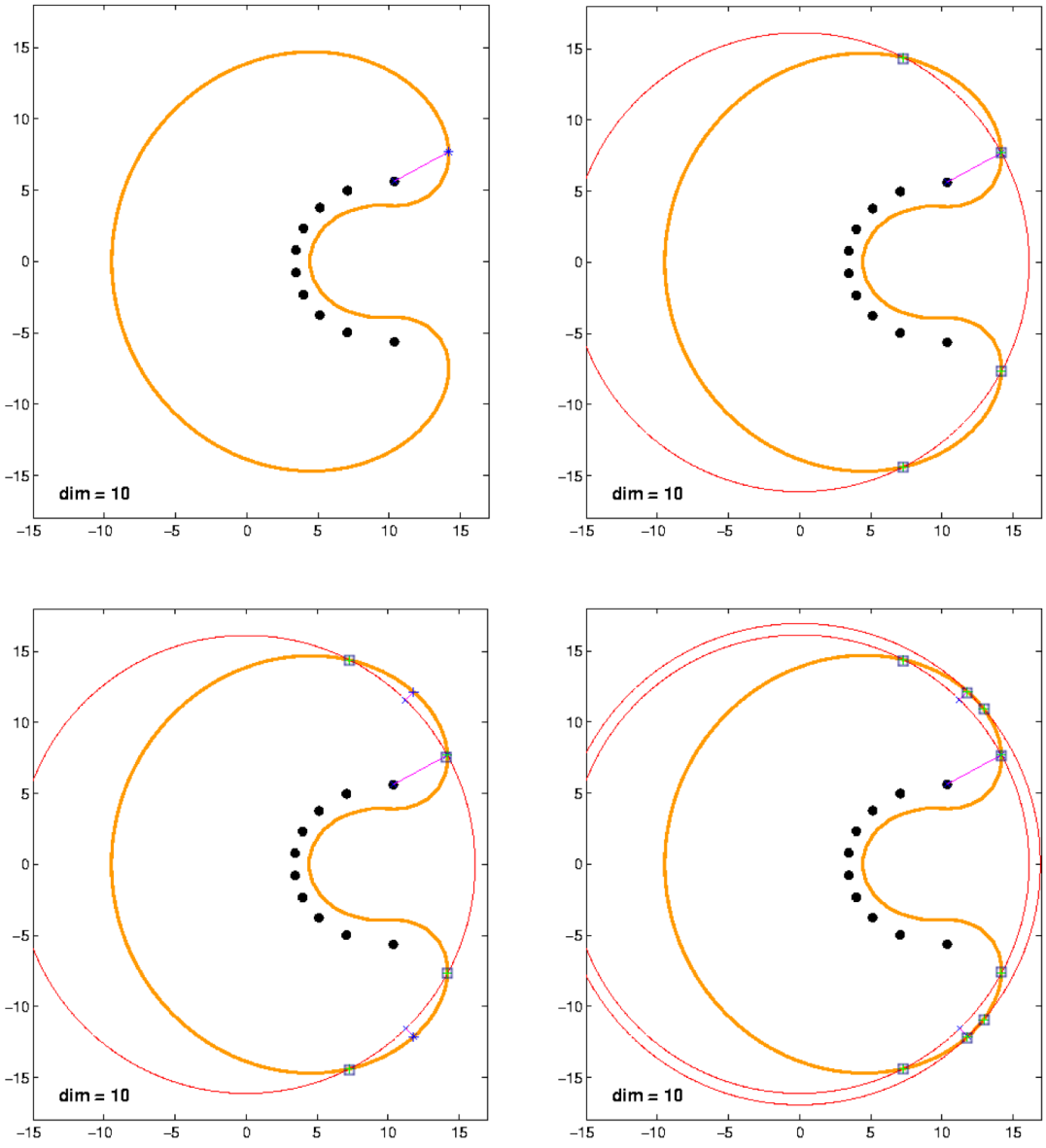

Figure 3.3: First two iterations of the pseudospectral radius algorithm on a shifted companion matrix.

Algorithm 3 Radial-circular algorithm for the pseudospectral radius

Call:	$\hat{\rho}_\epsilon \leftarrow$ `pspr`(A,ϵ,tol).
Input:	$A \in \mathbb{C}^{n\times n}$, $\epsilon \in \mathbb{R}_+$, $tol \in \mathbb{R}_+$ (tolerance for termination).
Output:	$\hat{\rho}_\epsilon \in \mathbb{R}_+$, the estimate value for the ϵ-pseudospectral radius.

Let θ_ρ be the angle of an eigenvalue with modulus $\rho(A)$, set $j = 0$, $\eta_0 = \rho(A)$ and $\Phi^0 = [\theta_\rho]$.
repeat

perform radial searches: Perform a radial search for each midpoint $\Phi_d^j \in \Phi^j$. Compute

$$\eta^{j+1} = \max\{\eta_\epsilon(\Phi_d^j) : \Phi_d^j \in \Phi^j\} \tag{3.13}$$

where η_ϵ is defined in (3.9).

perform circular search: Perform a circular search to find the intersection points of the circle with radius η^{j+1} and the ϵ-pseudospectrum boundary. Using these intersection points, determine the open intervals I_1^{j+1}, I_2^{j+1},..., $I_{m^{j+1}}^{j+1}$ in which $p(\eta^{j+1},\cdot)$ is negative. Compute the new set of midpoints

$$\Phi^{j+1} = \{\Phi_1^{j+1}, \Phi_2^{j+1}, \ldots, \Phi_{m^{j+1}}^{j+1}\},$$

where Φ_d^{j+1} is the midpoint of the interval I_d^{j+1},

$$\Phi_d^{j+1} = \begin{cases} \frac{\iota_d^{j+1}+\zeta_d^{j+1}}{2} & \text{if } \iota_d^{j+1} < \zeta_d^{j+1}, \\ \frac{\iota_d^{j+1}+\zeta_d^{j+1}+2\pi}{2} \bmod 2\pi & \text{otherwise} \end{cases}$$

increment j

until $\eta_j - \eta_{j-1} < tol$.
return η_j

It is possible to obtain a slight improvement in Algorithm 3 by changing the radial search to return the largest r in absolute value such that $p(r,\theta)=0$. Corollary 12 can be extended to show that the modulus of the pure imaginary eigenvalue of $K(\theta,\epsilon)$ with the largest imaginary part in absolute value is the largest zero of $p(\cdot,\theta)$ in absolute value. This version of the radial search may occasionally provide a better initial estimate; however, for the later iterations the gain is likely to be insignificant. To keep the description and analysis simple we use the definition (3.9).

In Algorithm 3 one point that is left unspecified is how the intervals $I_1^j, I_2^j, \ldots I_{m_j}^j$ can be determined from the intersection points returned by a circular search. One trivial and robust way is to sort the intersection points and compute $\sigma_{\min}(A-\eta^j e^{i\theta}I)$ at the midpoint θ of each adjacent pair. The adjacent pair constitutes an interval in which $p(\eta^j,\cdot)<0$ is satisfied if and only if $\sigma_{\min}(A-\eta^j e^{i\theta}I)<\epsilon$. Another possibility is to classify the intersection points as crossing or noncrossing zeros. We call the intersection point θ' a crossing zero of $p(r,\cdot)$ if $p(r,\cdot)$ has opposite sign on $(\theta'-\varepsilon,\theta']$ and $[\theta',\theta'+\varepsilon')$ for sufficiently small positive ε. Otherwise the intersection point is called a noncrossing zero of $p(r,\cdot)$. We can distinguish the crossing zeros from noncrossing zeros using the theorem below under the assumption that $\sigma_{\min}(A-re^{i\theta}I)$ is of multiplicity one for each intersection point $re^{i\theta}$.

Theorem 14 (Crossing versus Noncrossing Zeros during the Circular Searches). *Let $r\in\mathbb{R}^+$ and $e^{i\theta_0}$ be an eigenvalue of the pencil $P(r,\epsilon)-\lambda Q(r,\epsilon)$. Moreover, suppose that $\sigma_{\min}(A-re^{i\theta_0}I)$ is simple and equal to ϵ. Then θ_0 is a crossing zero of $p(r,\cdot)$ if and only if the algebraic multiplicity of the eigenvalue $e^{i\theta_0}$ is odd.*

Proof. By the definitions of $P(r,\epsilon)$ and $Q(r,\epsilon)$ (see (3.11))

$$P(r,\epsilon)-\lambda Q(r,\epsilon)=\det\begin{bmatrix} -\epsilon I & A-\lambda rI \\ rI-\lambda A^* & \lambda\epsilon I \end{bmatrix}.$$

We define the function $q:\mathbb{C}\to\mathbb{C}$ as the determinant of this matrix with the bottom block multiplied by $-\bar{\lambda}$,

$$q(\lambda)=(-1)^n\bar{\lambda}^n\det(P(r,\epsilon)-\lambda Q(r,\epsilon))=\det\begin{bmatrix} -\epsilon I & A-\lambda rI \\ |\lambda|^2A^*-\bar{\lambda}rI & -|\lambda|^2\epsilon I \end{bmatrix}. \tag{3.14}$$

Define a function $g:\mathbb{R}\to\mathbb{C}$ by $g(\theta)=q(e^{i\theta})$. Now if the multiplicity of $e^{i\theta_0}$ as the eigenvalue of the pencil $P(r,\epsilon)-\lambda Q(r,\epsilon)$ is m, we have

$$g(\theta)=q(e^{i\theta})=\beta(\theta)(e^{i\theta}-e^{i\theta_0})^m, \tag{3.15}$$

where $\beta : \mathbb{R} \to \mathbb{C}$ is a continuous function with $\beta(\theta_0) \neq 0$. Furthermore, when we make the substitution $\lambda = e^{i\theta}$ in the right-hand side of (3.14), we see that the eigenvalues of the resulting matrix are $\pm\sigma_j(A - re^{i\theta}I) - \epsilon$, *i.e.* plus and minus the singular values of $A - re^{i\theta}I$ decremented by ϵ. Therefore

$$g(\theta) = (-1)^n \prod_{j=1}^{n} (\sigma_j(A - re^{i\theta}I) - \epsilon)(\sigma_j(A - re^{i\theta}I) + \epsilon), \tag{3.16}$$

implying $g(\theta)$ is real valued for all θ.

Now for real small ε, we deduce from the equality

$$e^{i(\theta_0+\varepsilon)} - e^{i\theta_0} = e^{i\theta_0}(e^{i\varepsilon} - 1) = e^{i\theta_0} i\varepsilon + O(\varepsilon^2) = \varepsilon \left(e^{i(\theta_0+\pi/2)} + O(\varepsilon)\right)$$

and from (3.15) that

$$g(\theta_0 + \varepsilon) = \varepsilon^m (\beta(\theta_0 + \varepsilon) e^{mi(\theta_0+\pi/2)} + O(\varepsilon)) \equiv \varepsilon^m f(\theta_0, \varepsilon)$$

holds where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is continuous. Notice that because of the continuity of f and the fact that $f(\theta_0, 0) = \beta(\theta_0) e^{mi(\theta_0+\pi/2)}$ is a nonzero real number, $f(\theta_0, \varepsilon)$ and $f(\theta_0, -\varepsilon)$ are nonzero with the same sign. Therefore for all small $\hat{r}$, $g(\theta + \hat{r}) = \hat{r}^m f(\theta_0, \hat{r})$ and $g(\theta - \varepsilon) = (-\varepsilon)^m f(\theta_0, -\varepsilon)$ have different signs if and only if m is odd. But according to (3.16) the sign of $g(\theta)$ changes around θ_0 if and only if the sign of $p(r, \theta)$ changes. □

Theorem 14 allows us in principle to classify in which intervals $p(\eta^j, \cdot)$ is negative by evaluating $\sigma_{\min}(A - \eta^j e^{i\theta} I)$ only at the midpoint of one pair of intersection points computed in step 3, provided the assumption that $\sigma_{\min}(A - re^{i\theta}I)$ is simple at the intersection points is valid. In practice, however, evaluation of $\sigma_{\min}(A - \eta^j e^{i\theta} I)$ at every midpoint seems a simpler and more robust way to determine in which intervals $p(\eta^j, \cdot)$ is negative.

3.1.4 Convergence analysis

We claim that the sequence of iterates $\{\eta^j\}$ generated by Algorithm 3 converges to the ϵ-pseudospectral radius of A when $tol = 0$ so that the algorithm does not terminate. Recall that we assume the pencil for the circular searches is regular, which implies that there are at most $2n$ intersection points of the circle of radius r and the ϵ-pseudospectrum boundary. The convergence proof is analogous to that of the criss-cross method to compute the pseudospectral abscissa [12] (Theorem 3.2). Therefore we shall just give an outline of the proof.

First note that on a circle centered at the origin and with radius strictly between the spectral radius and the ϵ-pseudospectral radius, there are points

lying in the strict ϵ-pseudospectrum as shown by the following argument. Given a point z on the boundary of the ϵ-pseudospectrum, according to definition (1.16), $z \in \Lambda(A+E)$ for some E with norm ϵ. But the eigenvalues of $A+tE$ are continuous functions of $t \in [0,1]$. Therefore there must be a continuous path from each point on the ϵ-pseudospectrum boundary to an eigenvalue of A that, excluding the end point on the boundary, lies entirely in the strict ϵ-pseudospectrum.

If at some iteration j the ϵ-pseudospectral radius estimate η^j is equal to the ϵ-pseudospectral radius, there is nothing to prove. Thus suppose that none of the estimates is equal to the ϵ-pseudospectral radius. In this case the estimates $\{\eta^j\}$ are monotonically increasing, bounded above by the ϵ-pseudospectral radius and bounded below by the spectral radius. This can be easily shown by induction considering the update rule (3.13) and the definition of $\eta_\epsilon(\theta)$ in (3.9).

Since the estimates are in increasing order bounded above by the ϵ-pseudospectral radius, they must converge to a real number η^∞ less than or equal to the ϵ-pseudospectral radius. Suppose η^∞ is strictly less than the ϵ-pseudospectral radius. There must be open intervals such that the function $p(\eta^\infty, \theta)$ is non-positive. Otherwise we obtain a contradiction with the result stating that for all r between the spectral radius and the pseudospectral radius there are points lying inside the ϵ-pseudospectrum and on the circle centered at the origin with radius r. But from the existence of the open intervals in which the inequality $p(\eta^\infty, \theta) \leq 0$ is satisfied, it is possible to deduce $p(\eta^\infty, \Phi_k^j) \leq 0$ for sufficiently large j and for some k. Therefore the inequality $\eta^{j+1} \geq \eta^\infty$ holds for sufficiently large j. This contradicts the fact that the iterates are monotonically increasing. Therefore the limit η^∞ must be equal to the ϵ-pseudospectral radius. Thus we have the following theorem.

Theorem 15. *Suppose that the pencil $P(r,\epsilon) - \lambda Q(r,\epsilon)$ is regular for all positive r. Then the sequence $\{\eta^j\}$ generated by Algorithm 3 converges to $\rho_\epsilon(A)$.*

Just as in the criss-cross algorithm for the pseudospectral abscissa, we expect Algorithm 3 to converge to the pseudospectral radius quadratically under the same regularity assumption stated in [12], namely, that the global maximizers of the ϵ-pseudospectral radius problem (3.8) are regular. In [12] a point in the complex plane (x, y) is called regular if the multiplicity of the minimum singular value of $A - (x+iy)I$ is one and the pair of left and right singular vectors corresponding to this minimum singular value are not orthogonal to each other. To show the quadratic convergence, the approach in [12] (Section 4 and Section 5) can be followed. The crucial point that is worth noting here is that by Theorem 9 the function $p(r,\theta)$ is analytic whenever the minimum singular value of $A - re^{i\theta}I$ is positive and has multiplicity one. Additionally by Theorem 10, around a regular local maximum of the pseudospectral radius

problem the gradient of p must be a positive multiple of $(1, 0)$. Suppose that the point (r_0, θ_0) is a regular local maximum. Now an analogous argument to that of Theorem 4.1 and Corollary 4.5 in [12] applies to deduce the existence of a real-analytic function $f(\theta)$ near zero such that $p(r, \theta)$ and $r - r_0 + f(\theta - \theta_0)$ have the same signs for all r and θ sufficiently close to (r_0, θ_0). Moreover, the function f satisfies the properties

$$f(0) = f'(0) = \ldots = f^{(2k-1)}(0) = 0\,,\ f^{(2k)}(0) > 0 \tag{3.17}$$

for some $k \geq 1$. According to Section 5 in [12], since the pseudospectrum around a local maximum can be described by a function satisfying (3.17), Algorithm 3 converges quadratically to the global maximum, which is the pseudospectral radius in our case.

As argued in [12], generically (with probability one over the space of matrices) the multiplicity of $\sigma_{\min}(A - re^{i\theta}I)$ is one at the maximizer (r, θ). If the multiplicity of the minimum singular value is greater than one at a maximizer, the quadratic convergence proof outlined above does not apply, although it may be possible to extend the proof to cover such cases.

3.1.5 Singular pencils in the circular search

We first consider the geometrical interpretation of the singularity of the pencil in a circular search. When the boundary of the ϵ-pseudospectrum of A contains an arc of the circle of radius r centered at the origin, we infer from Theorem 13 that the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$ is singular. Notice that the reverse implication does not necessarily hold. For generic matrices the minimum singular value of $A - re^{i\theta}I$ has multiplicity one for all θ (see [12]) and Theorem 16 tells us that there are actually only two possibilities when the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$ is singular.

Theorem 16 (**Singular Pencils and Circular Pseudospectra**). *Given a positive real number r, suppose that the pencil $P(r, \epsilon) - \lambda Q(r, \epsilon)$ is singular and that $\sigma_{\min}(A - re^{i\theta}I)$ has multiplicity one for all $\theta \in [0, 2\pi)$. Then either*

- *the circle with radius r lies completely inside the strict ϵ-pseudospectrum, or*
- *the ϵ-pseudospectrum boundary contains the circle of radius r.*

Proof. By Theorem 13 the singularity of the pencil guarantees that, given an arbitrary $\theta \in [0, 2\pi)$, the matrix $A - re^{i\theta}I$ has ϵ as one of its singular values, so $p(r, \theta) \leq 0$. If for all θ, $p(r, \theta) < 0$ is satisfied, the first case of the theorem holds. So assume that there is a zero of $p(r, \cdot)$. By way of contradiction, suppose

that there exists $\tilde{\theta}$ such that $p(r,\tilde{\theta}) < 0$. Let $\hat{\theta}$ be the zero of $p(r,\cdot)$ closest to $\tilde{\theta}$. Without loss of generality, assume $\hat{\theta}$ is greater than $\tilde{\theta}$. For all $\theta \in [\tilde{\theta},\hat{\theta})$, $p(r,\theta) < 0$, so the smallest singular value of $A - re^{i\theta}I$ is strictly less than ϵ, and hence the second smallest singular value of $A - re^{i\theta}I$ is less than or equal to ϵ. It follows by the continuity of singular values that the second smallest singular value of $A - re^{i\hat{\theta}}I$ is less than or equal to ϵ. This contradicts the fact that $\sigma_{\min}(A - re^{i\hat{\theta}}I)$ is equal to ϵ and has multiplicity one. Thus $p(r,\theta) = 0$ for all θ, so the second case holds. □

Now returning to Algorithm 3, we note that for all j there is a zero of the function $p(\eta^j,\cdot)$ because of the way we update the estimates of the pseudospectral radius (3.13). Therefore, the circle of radius η^j cannot completely lie inside the strict pseudospectrum. In other words, for generic matrices the singularity of the pencil used by Algorithm 3 for the circular search implies that the ϵ-pseudospectrum boundary contains a circle.

In general, the presence of singular pencils is not desirable for Algorithm 3, because it is difficult to determine the singularity of a pencil. Thus our strategy to handle singular pencils is to try to avoid them. This turns out to be surprisingly simple. The next result is a corollary of Theorem 13.

Theorem 17 (**Avoiding Singular Pencils**). *Let r be a positive real number such that $\sigma_{\min}(A - re^{i\theta}I) > \epsilon$ for some θ. Then the pencil $P(r,\epsilon) - \lambda Q(r,\epsilon)$ is regular.*

For all r greater than η^1, by (3.9) $\sigma_{\min}(A - re^{i\theta_\rho}I) > \epsilon$. Therefore, as long as the initial estimate computed in floating point arithmetic $\hat{\eta}^1$ is greater than the exact initial estimate η^1, no singular pencils will be encountered. In particular, the convergence analysis of the previous subsection is valid. Trouble may occur, however, when $\hat{\eta}^1 < \eta^1$, in which case there may not exist θ such that $\sigma_{\min}(A - re^{i\theta}I) > \epsilon$. In general, when $\sigma_{\min}(A - \hat{\eta}^1 e^{i\theta}I) < \epsilon$ for all θ, the circle of radius $\hat{\eta}^1$ lies completely inside the ϵ-pseudospectrum, so the circular search in floating point arithmetic may potentially fail to return any intersection point.

All this discussion suggests raising the initial estimate $\hat{\eta}^1$ by a tolerance. In the next subsection we show that, provided structure-preserving backward stable eigenvalue solvers are used for the radial searches, $\hat{\eta}^1$ is the imaginary part of the largest imaginary eigenvalue of $K(\theta_\rho, \epsilon+\beta)$, where $\beta = O(\delta_{\text{mach}}(\|A\|+\epsilon))$. We have (see Theorem 11 and Corollary 12)

$$\sigma_{\min}(A - \hat{\eta}^1 e^{i\theta_\rho}I) = \epsilon + \beta.$$

We essentially want to increment $\hat{\eta}^1$ by a value δr such that

$$\sigma_{\min}(A - (\hat{\eta}^1 + \delta r)e^{i\theta_\rho}I) > \epsilon. \tag{3.18}$$

In a numerical implementation of Algorithm 3, the case we need to worry about is when β is negative. Assuming that the multiplicity of $\sigma_{\min}(A-\hat{\eta}^1 e^{i\theta_\rho} I)$ is one (so that Theorem 9 implies that $\sigma_{\min}(A - re^{i\theta_\rho} I)$ is real-analytic at $r = \hat{\eta}^1$), it follows from the equality

$$(\epsilon + \beta) + \delta r \frac{\partial \sigma_{\min}(A - re^{i\theta_\rho} I)}{\partial r}|_{r=\hat{\eta}^1} + O(\delta r^2) = \sigma_{\min}(A - (\hat{\eta}^1 + \delta r)e^{i\theta_\rho} I)$$

that for $\delta r = -\beta / \frac{\partial \sigma_{\min}(A - re^{i\theta_\rho} I)}{\partial r}|_{r=\hat{\eta}^1}$, $\sigma_{\min}(A-(\hat{\eta}^1+\delta r)e^{i\theta_\rho} I) = \epsilon + O(\delta r^2)$ holds. Since according to Theorem 9, $\frac{\partial \sigma_{\min}(A - re^{i\theta_\rho} I)}{\partial r}|_{r=\hat{\eta}^1} = \frac{\partial p(r,\theta_\rho)}{\partial r}|_{r=\hat{\eta}^1} = -\mathrm{Re}\ e^{i\theta_\rho} u^* v$ where u and v are unit left and right singular vectors corresponding to the minimum singular value of $A - \hat{\eta}^1 e^{i\theta_\rho} I$, we keep incrementing $\hat{\eta}^1$ by $\frac{\beta}{\mathrm{Re}\ e^{i\theta_\rho} u^* v}$ until (3.18) is satisfied. Usually it is sufficient to iterate once or twice to obtain a satisfactory δr.

3.1.6 Accuracy

We analyze the error introduced by a numerical implementation of Algorithm 3 with $tol = 0$ that generates increasing estimates in floating point arithmetic and terminates when the circular search fails to return any intersection point. The pseudospectral radius problem (3.8) may be ill-conditioned. This is the case when the pseudospectral radius is differentiable and the smallest left and right singular vectors at the global maximizer are close to being orthogonal (see Theorem 10). Therefore we focus on the backward error.

We start with an error analysis for the radial search. From Corollary 12 we know that the exact value $\eta_\epsilon(\theta) = r_\epsilon(\theta)$, where $r_\epsilon(\theta)i$ is the imaginary eigenvalue of $K(\theta, \epsilon)$ with the largest imaginary part. On the other hand, assuming that the eigenvalues are computed by a backward stable algorithm, the counterpart of $r_\epsilon(\theta)$ in floating point arithmetic, say $\hat{r}_\epsilon(\theta)$, is the largest imaginary part of the imaginary eigenvalues of a perturbed matrix

$$\tilde{K}(\theta, \epsilon) = K(\theta, \epsilon) + E, \tag{3.19}$$

where $\|E\| = O(\delta_{\mathrm{mach}} \|K(\theta, \epsilon)\|)$ or, since $\|K(\theta, \epsilon)\| \leq 2(\|A\| + \epsilon)$, $\|E\| = O(\delta_{\mathrm{mach}}(\|A\| + \epsilon))$. Additionally, when the algorithm used to solve the Hamiltonian eigenvalue problem is structure-preserving, the matrices E and $\tilde{K}(\theta, \epsilon)$ are Hamiltonian. The analysis for the radial search is valid only when a structure-preserving, backward stable Hamiltonian eigenvalue solver (see §A.1 for discussions on structure-preserving, backward stable Hamiltonian eigenvalue solvers) is used within Algorithm 3.

We first derive an upper bound on the result returned by the radial search in terms of the radius of nearby pseudospectra. The following result inspired

by [15] relates the eigenvalues of $\tilde{K}(\theta,\epsilon)$ and the $(\epsilon+\beta)$-pseudospectrum of A, where β is some real number with $|\beta|$ at most the norm of the perturbation matrix $\|E\|$.

Theorem 18 (**Accuracy of the Radial Search**). *Suppose that the Hamiltonian matrix $\tilde{K}(\theta,\epsilon)$ has the imaginary eigenvalue ir. Then $ir \in \Lambda(K(\theta,\epsilon+\beta))$ for some real β such that $|\beta| \leq \|E\|$.*

Proof. Since $ir \in \Lambda(\tilde{K}(\theta,\epsilon))$,

$$\det(\tilde{K}(\theta,\epsilon) - irI) = \det(J\tilde{K}(\theta,\epsilon) - irJ) = 0.$$

Notice that $J\tilde{K}(\theta,\epsilon) - irJ$ is Hermitian, meaning that the perturbed Hermitian matrix $J\tilde{K}(\theta,\epsilon) - irJ - JE = JK(\theta,\epsilon) - irJ$ has a real eigenvalue β which is at most $\|E\|$ in absolute value (from Weyl's Theorem; see for example [38], Theorem (4.3.1)). Now by the definition of $K(\theta,\epsilon)$ (see (3.10))

$$0 = \det(JK(\theta,\epsilon) - irJ - \beta I) = \det(K(\theta,\epsilon) + \beta J - irI) = \det(K(\theta,\epsilon+\beta) - irI).$$

Hence ir is an eigenvalue of $K(\theta,\epsilon+\beta)$. □

An immediate consequence of Theorem 18 is that $\hat{r}_\epsilon(\theta) \leq \eta_{\epsilon+\beta}(\theta)$ for some β with $|\beta| \leq \|E\|$; therefore the result of the radial search in floating point arithmetic, $\hat{r}_\epsilon(\theta)$, satisfies the inequality

$$\hat{r}_\epsilon(\theta) \leq \rho_{\epsilon+\|E\|}(A). \tag{3.20}$$

We now turn our attention to the circular search. In order to find the intersection points of the circle of radius r and the ϵ-pseudospectrum boundary, we compute the eigenvalues of the pencil $P(r,\epsilon) - \lambda Q(r,\epsilon)$. In floating point arithmetic, assuming a backward stable algorithm is used, we retrieve the eigenvalues of a nearby pencil $\tilde{P}(r,\epsilon) - \lambda\tilde{Q}(r,\epsilon)$. Additionally, for any nonnegative real μ, we make use of the notation

$$\tilde{P}(r,\mu) = P(r,\mu) + E_1 \quad \text{and} \quad \tilde{Q}(r,\mu) = Q(r,\mu) + E_2$$

where $E_1 = \tilde{P}(r,\epsilon) - P(r,\epsilon)$ and $E_2 = \tilde{Q}(r,\epsilon) - Q(r,\epsilon)$. The fact that the eigenvalue solver is backward stable implies that $\|E_1\| = O(\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A)))$ and $\|E_2\| = O(\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A)))$, since $\|E_1\| = O(\delta_{\text{mach}}\|P(r,\epsilon)\|)$ and $\|P(r,\epsilon)\| \leq (\|A\| + \epsilon + \rho_\epsilon(A))$, and similarly for $\|E_2\|$. The error analysis for the circular search involves the unitary matrix $D(\theta)$ (see (3.12)). The role of $D(\theta)$ in the analysis below is analogous to the role of J in the error analysis for the radial search in the sense that $D(\theta)(P(r,\epsilon) - e^{i\theta}Q(r,\epsilon))$ is Hermitian for all θ. In addition to the backward stability requirement on the generalized

eigenvalue solver, we also assume that it preserves the structure so that for all θ, $D(\theta)(\tilde{P}(r,\epsilon) - e^{i\theta}\tilde{Q}(r,\epsilon))$ is Hermitian. Unfortunately, we are not aware, at the moment, of the existence of a backward stable algorithm preserving this structure, but the assumption that the eigenvalue solver preserves this structure is essential for the analysis.

We are interested in bounding the estimate for the pseudospectral radius from below in terms of a nearby pseudospectral radius when the circular search does not return any intersection point. In this case the pencil $\tilde{P}(r,\epsilon) - \lambda\tilde{Q}(r,\epsilon)$ does not have any unit eigenvalue.

Theorem 19 (**Accuracy when the Circular Search Fails**). *Suppose that the pencil $\tilde{P}(r,\epsilon) - \lambda\tilde{Q}(r,\epsilon)$ does not have any unit eigenvalue and there exists θ such that $\sigma_{\min}(A - re^{i\theta}I) \geq \epsilon + \|E_1\| + \|E_2\|$. Then the pencil $P(r,\mu) - \lambda Q(r,\mu)$ does not have any unit eigenvalue for all positive $\mu \leq \epsilon - \|E_1\| - \|E_2\|$.*

Proof. Let $\chi_j(\theta)$ and $\varphi_j(\theta)$, $j = 1 \ldots 2n$, denote the eigenvalues of $N(\theta) = D(\theta)(\tilde{P}(r,0) - e^{i\theta}\tilde{Q}(r,0))$ and $R(\theta) = D(\theta)(P(r,0) - e^{i\theta}Q(r,0))$ as functions of θ in descending order. Notice that $\chi_j(\theta)$ and $\varphi_j(\theta)$ are real-valued continuous functions of θ, since the entries of the matrices $N(\theta)$ and $R(\theta)$ are continuous with respect to θ and both of the matrices are Hermitian for all θ. Note also that

$$|\chi_j(\theta) - \varphi_j(\theta)| \leq \|E_1\| + \|E_2\| \tag{3.21}$$

holds for all j and θ. This inequality follows from the fact that $\|N(\theta) - R(\theta)\| = \|E_1 - e^{i\theta}E_2\| \leq \|E_1\| + \|E_2\|$, so the corresponding eigenvalues of the Hermitian matrices cannot differ by more than $\|E_1\| + \|E_2\|$.

Since the pencil $\tilde{P}(r,\epsilon) - \lambda\tilde{Q}(r,\epsilon)$ does not have any unit eigenvalue, for all θ

$$\begin{aligned}
\det(D(\theta)(\tilde{P}(r,\epsilon) - e^{i\theta}\tilde{Q}(r,\epsilon))) &= \det(D(\theta)(\tilde{P}(r,0) - e^{i\theta}\tilde{Q}(r,0) - \epsilon D^*(\theta))) \\
&= \det(D(\theta)(\tilde{P}(r,0) - e^{i\theta}\tilde{Q}(r,0)) - \epsilon I) \\
&= \det(N(\theta) - \epsilon I) \\
&\neq 0.
\end{aligned}$$

Hence the function $\chi_j(\theta) \neq \epsilon$ for all j and θ. But the assumption that $\sigma_{\min}(A - re^{i\hat{\theta}}I) \geq \epsilon + \|E_1\| + \|E_2\|$ for some $\hat{\theta}$ implies that, for all $1 \leq j \leq n$,

$$\varphi_j(\hat{\theta}) \geq \epsilon + \|E_1\| + \|E_2\|, \tag{3.22}$$

since for all θ the eigenvalues of $R(\theta)$ consist of plus and minus the singular values of $A - re^{i\theta}I$. When we combine (3.21) and (3.22), we see that for all $1 \leq j \leq n$,

$$\chi_j(\hat{\theta}) \geq \epsilon. \tag{3.23}$$

Now by way of contradiction, suppose that there exists a positive $\mu \leq \epsilon - \|E_1\| - \|E_2\|$ such that the pencil $P(r,\mu) - \lambda Q(r,\mu)$ has a unit eigenvalue, say $e^{i\tilde{\theta}}$. Then

$$\det(D(\tilde{\theta})(P(r,\mu) - e^{i\tilde{\theta}}Q(r,\mu))) = \det(D(\tilde{\theta})(P(r,0) - e^{i\tilde{\theta}}Q(r,0)) - \mu I) = 0$$

meaning that for some $j \leq n$, $\varphi_j(\tilde{\theta}) = \mu$, since μ is positive. It follows from (3.21) that $\chi_j(\tilde{\theta}) \leq \mu + \|E_1\| + \|E_2\| \leq \epsilon$ is satisfied. But for the same j, (3.23) holds as well. We conclude from the intermediate value theorem that there exists θ' satisfying $\chi_j(\theta') = \epsilon$. This contradicts the fact that $\chi_j(\theta) \neq \epsilon$ for all θ and j. □

In exact arithmetic the circular search fails when the circle of radius r lies either completely inside or completely outside the pseudospectrum. In the theorem above, we need the condition that there exists a point $re^{i\theta}$ on the circle of radius r such that $\sigma_{\min}(A - re^{i\theta}I) \geq \epsilon + \|E_1\| + \|E_2\|$ in order to distinguish these two cases. When $\sigma_{\min}(A - re^{i\theta}I) \geq \epsilon$ and the derivative of $\sigma_{\min}(A - r'e^{i\theta'}I)$ with respect to θ' at (r,θ) is not very small, such a point exists on the circle of radius r in a small neighborhood of θ. In the previous subsection we discussed how to generate estimates r such that $\sigma_{\min}(A - re^{i\theta_\rho}I) \geq \epsilon$.

Focusing on the implications of Theorem 19, whenever the circular search in floating point arithmetic fails for some $r > \rho(A)$ and there exists a point (r,θ) with $\sigma_{\min}(A - re^{i\theta}I) \geq \epsilon + \|E_1\| + \|E_2\|$, then for all $\tau \geq \|E_1\| + \|E_2\|$, the $(\epsilon - \tau)$-pseudospectrum lies inside the circle of radius r. We accordingly infer the lower bound

$$\rho_{\epsilon - \|E_1\| - \|E_2\|}(A) \leq r. \tag{3.24}$$

Now we are ready to find the backward error of the algorithm. First, since the estimates are increasing in floating point arithmetic, the algorithm is guaranteed to terminate. At the termination, the estimate value must satisfy the upper bound (3.20), because it is generated by a radial search at the previous iteration. Moreover, at the last iteration the circular search fails, meaning that the lower bound (3.24) on the final estimate holds as well. Combining these bounds and from the continuity of $\rho_\epsilon(A)$ with respect to ϵ (see Theorem 7), we see that the estimate $\hat{\rho}_\epsilon(A)$ at the termination satisfies

$$\hat{\rho}_\epsilon(A) = \rho_{\epsilon+\beta}(A)$$

where $\beta = O(\delta_{\text{mach}}(\|A\| + \epsilon + \rho_\epsilon(A)))$, *i.e.* the final estimate is the solution of a nearby pseudospectral radius problem for the same matrix.

Our analysis above depends on the usage of the proper eigenvalue solvers. Possible choices for the structured eigenvalue solvers are provided in Appendix A.

3.1.7 The pseudospectral radius of a matrix polynomial

The ϵ-pseudospectral radius of P is the largest of the moduli of the points in the ϵ-pseudospectrum of P,

$$\rho_\epsilon(P,\gamma) = \max\{|z| : z \in \Lambda_\epsilon(P,\gamma)\} \tag{3.25}$$

or, using the characterization (1.15),

$$\rho_\epsilon(P,\gamma) = \max\{|z| : \sigma_{\min}\left(\frac{P(z)}{p_\gamma(|z|)}\right) \leq \epsilon\}. \tag{3.26}$$

The ϵ-pseudospectral radius of a polynomial can be computed by means of Algorithm 3, but we need to clarify how to do a circular search and a radial search on the ϵ-pseudospectrum of a polynomial.

An efficient procedure to find the intersection points of the boundary of the ϵ-pseudospectrum and a circle of given radius centered at the origin requires solving a $*$-palindromic eigenvalue problem of double size and double degree (see §A.4 for the discussions on $*$-palindromic eigenvalue problems).

Theorem 20 (**Circular Search on the Polynomial ϵ-pseudospectrum**). *For given positive real numbers r and ϵ, the matrix $\frac{P(re^{i\theta})}{p_\gamma(r)}$ has ϵ as one of its singular values if and only if the matrix polynomial $\mathcal{P}(r,\epsilon) = \sum_{l=0}^{2k} \lambda^l \mathcal{P}_l(r,\epsilon)$ has the eigenvalue $e^{i\theta/2}$ where for $l \neq k$*

$$\mathcal{P}_l(r,\epsilon) = \begin{cases} \begin{bmatrix} 0 & K_{l/2} r^{l/2} \\ K^*_{k-l/2} r^{k-l/2} & 0 \end{bmatrix} & \textit{if } l \textit{ is even} \\ 0 \quad \textit{if } l \textit{ is odd,} \end{cases}$$

and

$$\mathcal{P}_k(r,\epsilon) = \begin{cases} \begin{bmatrix} -\epsilon p_\gamma(r) I & K_{k/2} r^{k/2} \\ K^*_{k/2} r^{k/2} & -\epsilon p_\gamma(r) I \end{bmatrix} & \textit{if } k \textit{ is even} \\ \begin{bmatrix} -\epsilon p_\gamma(r) I & 0 \\ 0 & -\epsilon p_\gamma(r) I \end{bmatrix} & \textit{if } k \textit{ is odd.} \end{cases}$$

Proof. The scalar $\epsilon p_\gamma(r)$ is a singular value of the matrix $P(re^{i\theta})$ if and only if the Hermitian matrix

$$\begin{bmatrix} -\epsilon p_\gamma(r) I & P(re^{i\theta}) \\ (P(re^{i\theta}))^* & -\epsilon p_\gamma(r) I \end{bmatrix}$$

is singular or, by multiplying the leftmost and lower blocks by $e^{ik\theta/2}$,

$$\begin{bmatrix} -\epsilon p_\gamma(r)e^{ik\theta/2}I & P(re^{i\theta}) \\ e^{ik\theta}(P(re^{i\theta}))^* & -\epsilon p_\gamma(r)e^{ik\theta/2} \end{bmatrix} = \sum_{l=0}^{2k} e^{il\theta/2}\mathcal{P}_l(r,\epsilon)$$

is singular, implying that $e^{i\theta/2}$ is an eigenvalue of $\mathcal{P}(r,\epsilon)$. □

The unit eigenvalues of $\mathcal{P}(r,\epsilon)$ provide us a superset of the intersection points of the ϵ-pseudospectrum with the circle of radius r. The point $re^{i\theta}$ is an intersection point if and only if $e^{i\theta/2} \in \Lambda(\mathcal{P}(r,\epsilon))$ and the equality $\sigma_{\min}\left(\frac{P(re^{i\theta})}{p_\gamma(r)}\right) = \epsilon$ holds. Remarkably $(\mathcal{P}_l(r,\epsilon))^* = \mathcal{P}_{2k-l}(r,\epsilon)$, meaning the matrix polynomial $\mathcal{P}(r,\epsilon)$ is $*$-palindromic with the unit eigenvalues or the eigenvalues in pairs $(\lambda, 1/\bar{\lambda})$. We describe how to extract the unit eigenvalues of such matrix polynomials in §A.4.

The radial searches are performed by solving a $*$-even polynomial eigenvalue problem. Recall that when $\gamma_k > 0$ and $\sigma_{\min}\left(\frac{K_k}{\gamma_k}\right) \leq \epsilon$, the ϵ-pseudospectrum of P is unbounded. For the radial searches we require the ϵ-pseudospectrum to be bounded.

Theorem 21 (**Radial Search on the Polynomial ϵ-pseudospectrum**). *Let $\theta \in [0, 2\pi)$ and ϵ be a positive real scalar such that $\sigma_{\min}\left(\frac{K_k}{\gamma_k}\right) > \epsilon$. Then the largest r such that $\sigma_{\min}\left(\frac{P(re^{i\theta})}{p_\gamma(r)}\right) = \epsilon$ is the imaginary part of the upper-most imaginary eigenvalue of the matrix polynomial $\mathcal{K}(\theta,\epsilon) = \sum_{j=0}^{k} \lambda^j \mathcal{K}_l(\theta,\epsilon)$ with*

$$\mathcal{K}_0(\theta,\epsilon) = \begin{bmatrix} -\epsilon\gamma_0^2 I & K_0^* \\ K_0 & -\epsilon I \end{bmatrix},$$

for $l > 0$ when l is odd,

$$\begin{aligned} \mathcal{K}_l(\theta,\delta) &= \begin{bmatrix} 0 & (-1)^{(l+1)/2} i K_l^* e^{-il\theta} \\ (-1)^{(l+1)/2} i K_l e^{il\theta} & 0 \end{bmatrix} && 1 \leq l \leq k \\ \mathcal{K}_l(\theta,\delta) &= 0 && k+1 \leq l < 2k, \end{aligned}$$

and, when l is even,

$$\begin{aligned} \mathcal{K}_l(\theta,\delta) &= \begin{bmatrix} (-1)^{l/2+1}\epsilon\gamma_{l/2}^2 I & (-1)^{l/2} K_l^* e^{-il\theta} \\ (-1)^{l/2} K_l e^{il\theta} & 0 \end{bmatrix} && 1 \leq l \leq k \\ \mathcal{K}_l(\theta,\delta) &= \begin{bmatrix} (-1)^{l/2+1}\epsilon\gamma_{l/2}^2 I & 0 \\ 0 & 0 \end{bmatrix} && k+1 \leq l \leq 2k. \end{aligned}$$

Proof. The polynomial P can also be written as

$$P(re^{i\theta}) = \sum_{j=0}^{k} (ri)^j ((-i)^j e^{ij\theta} K_j).$$

Therefore by making the substitution $x = 0$ and replacing K_j by $(-i)^j e^{ij\theta} K_j$ in Theorem 5, we deduce that the set of r such that $\frac{P(re^{i\theta})}{p_\gamma(r)}$ has ϵ as one of its singular values is the same as the set of r such that ri is an eigenvalue of $\mathcal{K}(\theta, \epsilon)$. Let r_* be the largest r such that ri is an eigenvalue of $\mathcal{K}(\theta, \epsilon)$. Since the smallest singular value of $\frac{P(re^{i\theta})}{p_\gamma(r)}$ is a continuous function of r and as r goes to infinity in the limit it approaches $\sigma_{\min}\left(\frac{K_k}{\gamma_k}\right) > \epsilon$, the strict inequality

$$\sigma_{\min}\left(\frac{P(r_* e^{i\theta})}{p_\gamma(r_*)}\right) < \epsilon$$

cannot be satisfied, so the result follows. □

3.2 Numerical radius

For a matrix A the modulus of the point furthest away from the origin in the field of values

$$r(A) = \max\{|w| : w \in F(A)\} \tag{3.27}$$

is called the *numerical radius*. Figure 3.4 shows the point where the numerical radius of the normally distributed matrix (whose pseudospectrum is displayed in Figure 3.1) is attained on the field of values. Its benefit in analyzing the first-order discrete autonomous system is revealed by the upper bound

$$\|A^k\| \leq 2r(A)^k. \tag{3.28}$$

on the norm of the powers of A. This is an immediate consequence of the lower bound [39]

$$\frac{\|A\|}{2} \leq r(A) \tag{3.29}$$

together with the power inequality $r(A^k) \leq r(A)^k$ [64]. The bound (3.28) ensures that if the numerical radius of A is small, the initial growth does not happen or else it is insignificant. The quantity $r(A)$ captures the norm of A as well as the asymptotic behavior of the discrete first-order autonomous system. Therefore it is a desirable measure for the analysis of the classical iterative systems for which the error can be represented by first-order recurrences. The

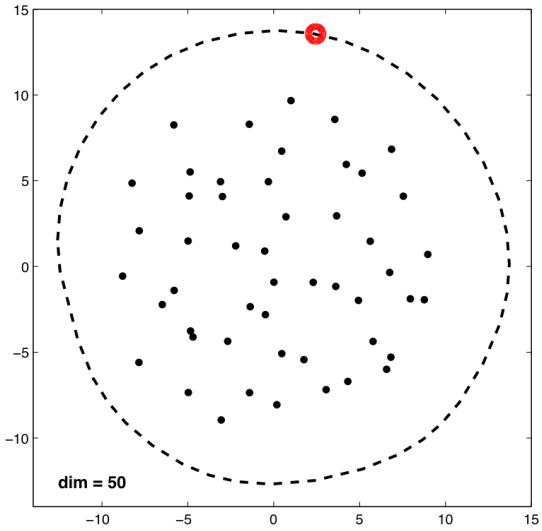

Figure 3.4: The field of values of the matrix for which the pseudospectra are illustrated in Figure 3.1. A circle marks the point where the numerical radius is attained.

analysis of the classical iterative methods using the field of values and the numerical radius has been studied by Axelsson *et.al.* [2] and Eiermann [23].

A measure analogous to the numerical radius for continuous-time dynamical systems is the numerical abscissa, the real component of the rightmost point in the field of values,

$$\alpha_F(A) = \max\{\text{Re } z : \ z \in F(A)\}.$$

Though intuitively computations of the numerical abscissa and the numerical radius seem equally difficult, the former can be reduced to an eigenvalue problem [39]:

$$\alpha_F(A) = \lambda_{\max}(H(A)), \tag{3.30}$$

where $H(A) = \frac{1}{2}(A + A^*)$. Multiplying A by $e^{i\theta}$ rotates the field of values of A by θ. Consequently, the numerical radius of A can be viewed as the global maximum of an optimization problem with a single real variable

$$r(A) = \max_{\theta \in [0,2\pi)} \alpha_F(Ae^{i\theta}). \tag{3.31}$$

Combining (3.30) and (3.31) yields the following characterization of the numerical radius

$$r(A) = \max_{\theta \in [0,2\pi)} \lambda_{\max}(H(Ae^{i\theta})). \tag{3.32}$$

For the computation of the numerical radius, the most recent method was suggested by He and Watson [34]. The method introduced in [34] is based on

finding a local maximum of the eigenvalue optimization problem (3.32) and verifying whether the local maximum is actually the global maximum by solving a generalized eigenvalue problem. However, the simple iteration introduced in [34] to locate a local maximum is not guaranteed to converge. Here we describe an algorithm that generates estimates converging to the numerical radius in exact arithmetic. The local convergence rate is usually quadratic. The algorithm is analogous to the Boyd-Balakrishnan algorithm for the $\mathbf{H}_\infty$ norm [9] and depends on the solution of the generalized eigenvalue problems used for checking whether a local maximum is the global maximum in [34].

Given the matrix A, let us define $f : [0, 2\pi) \to \mathbb{R}$ by

$$f(\theta) = \lambda_{\max}(H(Ae^{i\theta})). \tag{3.33}$$

Observe that for each $\theta \in [0, 2\pi)$, $f(\theta) \in [-\|A\|, \|A\|]$. Our aim is to find the global maximum of f.

In our algorithm, we need to determine θ values satisfying $f(\theta) = \hat{r}$, where $\hat{r} \geq 0$ is a numerical radius estimate. Consider the pencil $R(\hat{r}) - \lambda S$ with

$$R(\hat{r}) = \begin{bmatrix} 2\hat{r}I & -A^* \\ I & 0 \end{bmatrix}, \; S = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}.$$

In [34], it is proved that given a real number $\hat{r} \geq \min_\theta f(\theta)$, the pencil $R(\hat{r}) - \lambda S$ has an eigenvalue on the unit circle or is singular if and only if the inequality $\hat{r} \leq r(A)$ holds. Using the theorem in [34], we can decide whether there is a θ satisfying $f(\theta) = \hat{r}$; however, this theorem does not tell us what those θ values are. For this purpose we state a slightly modified version.

Theorem 22. *The pencil $R(\hat{r}) - \lambda S$ has the eigenvalue $e^{i\theta}$ or is singular if and only if the Hermitian matrix $H(Ae^{i\theta})$ has $\hat{r}$ as one of its eigenvalues.*

Proof. The equality $\det(R(\hat{r}) - e^{i\theta}S) = 0$ is satisfied if and only if the matrix

$$\begin{bmatrix} 2\hat{r}I - e^{i\theta}A & -A^* \\ I & -e^{i\theta}I \end{bmatrix}$$

is singular. Multiplying the bottom block row of this matrix by $e^{-i\theta}$, we see that this $2n \times 2n$ matrix is singular if and only if the $n \times n$ matrix $e^{i\theta}A + e^{-i\theta}A^* - 2\hat{r}I$ is singular. Therefore, the Hermitian matrix $H(Ae^{i\theta})$ has $\hat{r}$ as one of its eigenvalues if and only if the matrix $R(\hat{r}) - e^{i\theta}S$ is rank-deficient. □

From Theorem 22 it follows that, as long as the pencil $R(\hat{r}) - \lambda S$ is regular for a given $\hat{r}$, we can solve the generalized eigenvalue problem $R(\hat{r}) - \lambda S$ and extract the angles of the eigenvalues on the unit circle to obtain a superset of θ

values satisfying $f(\theta) = \hat{r}$. To determine the exact set, for each angle θ' that is extracted, the eigenvalues of $H(Ae^{i\theta'})$ need to be computed. Only those angles for which $H(Ae^{i\theta'})$ has $\hat{r}$ as the largest eigenvalue should be kept.

Now that we know how to compute the intersection points of a horizontal line with the graph of f efficiently, we suggest an iterative algorithm. At the jth iteration the algorithm generates an estimate of the numerical radius, r^j, and a set of open intervals, $I_1^j, I_2^j \ldots I_{m^j}^j$, where, as earlier, $I_{m_j}^j$ may wrap around the circle. The function f is greater than r^j in each interval I_l^j, $1 \leq l \leq m^j$ (*i.e.* for all $\theta \in I_l^j$, $f(\theta) > r^j$) and exactly r^j at the end points of the intervals. At the jth iteration the new estimate r^j is set to the maximum value attained by the function f at the midpoints of the open intervals produced at the previous iteration. Then the open intervals at the jth iteration are obtained using Theorem 22 followed by the maximum eigenvalue checks.

A robust way to determine $I_1^j, I_2^j, \ldots, I_{m^j}^j$ from the set of intersection points is to sort the intersection points and to compute f at the midpoint of each adjacent pair of points. The pencil $R(\hat{r}) - \lambda S$ is $*$-symplectic, so just as with circular searches in the previous section, this problem can be reduced to a Hamiltonian eigenvalue problem and the eigenvalue solver described in [7] can be applied (see Appendix A for the details). It is easy to avoid singular pencils, since for all $\hat{r}$ greater than the initial estimate $r^1 = f(0)$, the pencil $R(\hat{r}) - \lambda S$ is guaranteed to be regular. Note also that we can compute f accurately because of the fact that the eigenvalues of symmetric matrices are well-conditioned.

Algorithm 4 is an extension of the Boyd-Balakrishnan algorithm [9] to the numerical radius. Thus a similar convergence proof applies (based on the fact that the length of the greatest open interval is at least halved at each iteration). We believe that a proof along the line of the argument in [9] is applicable to show that the algorithm converges quadratically to the value $r(A)$ and that the accuracy analysis in the previous section for the pseudospectral radius can be extended to Algorithm 4.

3.3 Distance to instability

For discrete systems the boundary of the unstable region is the circumference of the unit circle. The distance to the closest unstable discrete system and the ϵ-pseudospectral radius are closely related. In particular,

$$\begin{aligned} \beta_d(A) < \epsilon &\iff \rho_\epsilon(A) < 1 \\ \beta_d(P, \gamma) < \epsilon &\iff \rho_\epsilon(P, \gamma) < 1 \end{aligned}$$

where β_d denotes the discrete distance to instability. We have observed in Figure 2.2 that perturbations with norm on the order of 10^{-7} are sufficient to move

Algorithm 4 Boyd-Balakrishnan type algorithm for the numerical radius

Call:	$\hat{r} \leftarrow$ `numrad`(A,tol).
Input:	$A \in \mathbb{C}^{n\times n}$, $tol \in \mathbb{R}_+$ (tolerance for termination).
Output:	$\hat{r} \in \mathbb{R}_+$, the estimate value for the numerical radius.

Set $j = 0$, $\phi^0 = [0]$ and $r^0 = 0$.
repeat

Update the numerical radius estimate: Compute r^{j+1} using the formula

$$r^{j+1} = \max\{f(\theta) : \theta \in \phi^j\}. \tag{3.34}$$

Update the set of the midpoints: Find θ values for which $f(\theta) = r^{j+1}$ holds. From these infer the open intervals I_l^{j+1}, for $l = 1, \ldots, m^{j+1}$ such that $\forall\theta \in I_l^{j+1}$, $f(\theta) > r^{j+1}$. Calculate the new set of midpoints

$$\phi^{j+1} = \{\phi_1^{j+1}, \phi_2^{j+1}, \ldots, \phi_{m^{j+1}}^{j+1}\}$$

where ϕ_l^{j+1} is the midpoint of the open interval $I_l^{j+1} = (\iota_l^{j+1}, \zeta_l^{j+1})$ (with the possible exception of $I_{m^{j+1}}^{j+1} = (\iota_{m^{j+1}}^{j+1}, 2\pi) \cup [0, \zeta_{m^{j+1}}^{j+1})$ in case the last interval wraps around)

$$\phi_l^{j+1} = \begin{cases} \frac{\iota_l^{j+1}+\zeta_l^{j+1}}{2} & \text{if } \iota_l^{j+1} < \zeta_l^{j+1}, \\ \frac{\iota_l^{j+1}+\zeta_l^{j+1}+2\pi}{2} \bmod 2\pi & \text{otherwise.} \end{cases}$$

Increment j.
until $r_j - r_{j-1} < tol$.
return r_j

the eigenvalues of the upper triangular matrix onto the unit circle and make it unstable. Indeed, its discrete distance to instability is 3.06×10^{-8}.

For a discrete system the distances to instability of the matrix A and the matrix polynomial P have the equivalent characterizations

$$\beta_d(A) = \inf_{\theta \in [0,2\pi)} \sigma_{\min}(A - e^{i\theta} I), \tag{3.35}$$

$$\beta_d(P, \gamma) = \inf_{\theta \in [0,2\pi)} \sigma_{\min}\left(\frac{P(e^{i\theta})}{p_\gamma(1)}\right). \tag{3.36}$$

Given an estimate $\hat{\beta}$ for the discrete distance to instability, the unit eigenvalues $e^{i\theta}$ of the $*$-symplectic pencil $P(1, \hat{\beta}) - \lambda Q(1, \hat{\beta})$ (defined by (3.11)) and the unit eigenvalues $e^{i\theta/2}$ of the $*$-palindromic matrix polynomial $\mathcal{P}(1, \hat{\beta})$ (defined in Theorem 20) provide us supersets of the θ values satisfying the equalities

$$\sigma_{\min}(A - e^{i\theta} I) = \hat{\beta},$$
$$\sigma_{\min}\left(\frac{P(e^{i\theta})}{p_\gamma(1)}\right) = \hat{\beta}.$$

Therefore Algorithm 2 can be used to compute $\beta_d(A)$ and $\beta_d(P, \gamma)$. Notice that for the discrete distance to instability of the matrix polynomial P, we can simply ignore the scaling γ as the polynomial $p_\gamma(|e^{i\theta}|) = p_\gamma(1) = \|\gamma\|$ is constant. This in turn implies that it does not matter which coefficients are allowed to vary.

3.4 Numerical examples

The first subsection below focuses on the ratio $\frac{\rho_\epsilon(A)-1}{\epsilon}$ as a function of ϵ for the upper triangular matrix of §1.3.1 and a scaled Grcar matrix. We concluded from the analysis in §3.1.6 that the algorithm for the ϵ-pseudospectral radius under rounding errors returns the $(\epsilon + \zeta)$-pseudospectral radius for some $\zeta = O(\epsilon_{\text{mach}}(\|A\| + \rho_\epsilon(A) + \epsilon))$. The same conclusion can be drawn about the accuracy of the radial search. The example in §3.4.2 illustrates that indeed the radial search can fail. §3.4.3 shows the variation in the running times of the algorithms as the size of the input matrices is increased. In §3.4.4 the extensions of the algorithms for matrix polynomials are run on sample examples. For all of the algorithms in this chapter we observe fast convergence in practice similar to the convergence of the algorithms in the previous chapter.

3.4.1 Bounding the discrete Kreiss constant

We have seen in the previous chapter that for the upper triangular matrix with the entries $a_{lj} = -0.3,\ (l \leq j)$ the norm of e^{At} decays monotonically, which is

consistent with the continuous Kreiss constant being equal to one and attained at ∞. We see the opposite picture when we consider the discrete Kreiss constant

$$\mathcal{K}_d(A) = \sup_{\epsilon>0} \frac{\rho_\epsilon(A) - 1}{\epsilon}. \tag{3.37}$$

In Figure 3.5 on the top we plot the ratio $\frac{\rho_\epsilon - 1}{\epsilon}$ as a function of ϵ. The ϵ-pseudospectral radius for various ϵ is computed for the upper triangular matrix by Algorithm 3. Unlike the continuous Kreiss constant, the discrete Kreiss constant is close to 6×10^5 and attained around $\epsilon = 10^{-7}$, a value that is slightly larger than the discrete distance to instability of the upper triangular matrix. Also for the Grcar matrix with entries equal to 0.4 on the diagonal, first, second and third superdiagonal and -0.4 on the subdiagonal, the similar plot in Figure 3.5 at the bottom indicates a transient peak that is not as big in magnitude as the transient peak of the upper triangular matrix. In both of the plots when $\rho_\epsilon < 1$, the ratio is replaced by zero.

3.4.2 Accuracy of the radial search

The analysis in §3.1.6 showed that in finite precision we perform radial searches on *nearby* pseudospectra. In particular, Theorem 18 implies that instead of the quantity $\eta_\epsilon(\theta)$ we retrieve $\eta_{\epsilon+\zeta}(\theta)$ under rounding errors, where $\zeta = O(\epsilon_{\text{mach}}(\|A\| + \rho_\epsilon(A) + \epsilon))$. The trouble in performing the radial search accurately is that $\eta_\epsilon(\theta)$ does not depend continuously on ϵ.

Consider the example in Figure 3.6, where the curve shown is the boundary of the ϵ-pseudospectrum for the 50×50 Demmel example (available in *EigTool* [73]), shifted by $2I$, for $\epsilon = 10^{-0.9537}$. The ϵ-pseudospectrum contains all of the points inside the outer closed curve except the points inside the inner closed curve. For the particular ϵ value chosen, the inner curve is almost tangent to the outer curve at $(x, y) = (0, 22.9399)$. However, when we take a closer look at that region (around $(0, 22.9399)$) in Figure 3.6 on the right, we see that the closed curves are actually disjoint.

In exact arithmetic the radial search in the direction $\theta = 0$ must return the modulus of the point on the outer closed curve. But the radial search in finite precision produces the modulus of a point on the inner curve as illustrated in Figure 3.7. The ray in the direction $\theta = 0$ intersects the ϵ-pseudospectrum boundary at three distinct points. Unfortunately, the value returned by the radial search is the smallest among the moduli of these three points which is equal to 1.6571, while the largest of the moduli is 22.9399. While this might seem alarming, observe that if we slightly decrease ϵ, the two closed curves merge and the ray in the direction $\theta = 0$ intersects the ϵ-pseudospectrum at

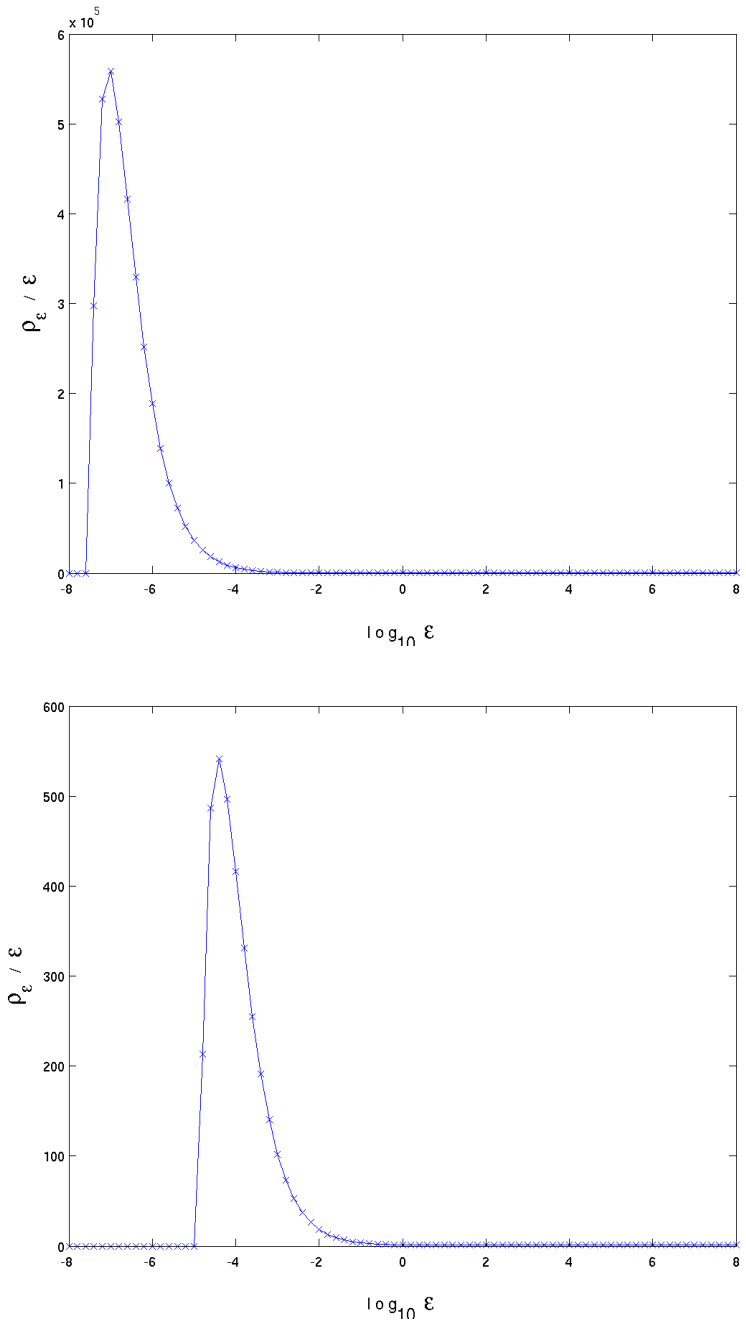

Figure 3.5: The ratio $(\rho_\epsilon - 1)/\epsilon$ is plotted as a function of ϵ for the upper triangular matrix on the top and for the Grcar matrix at the bottom.

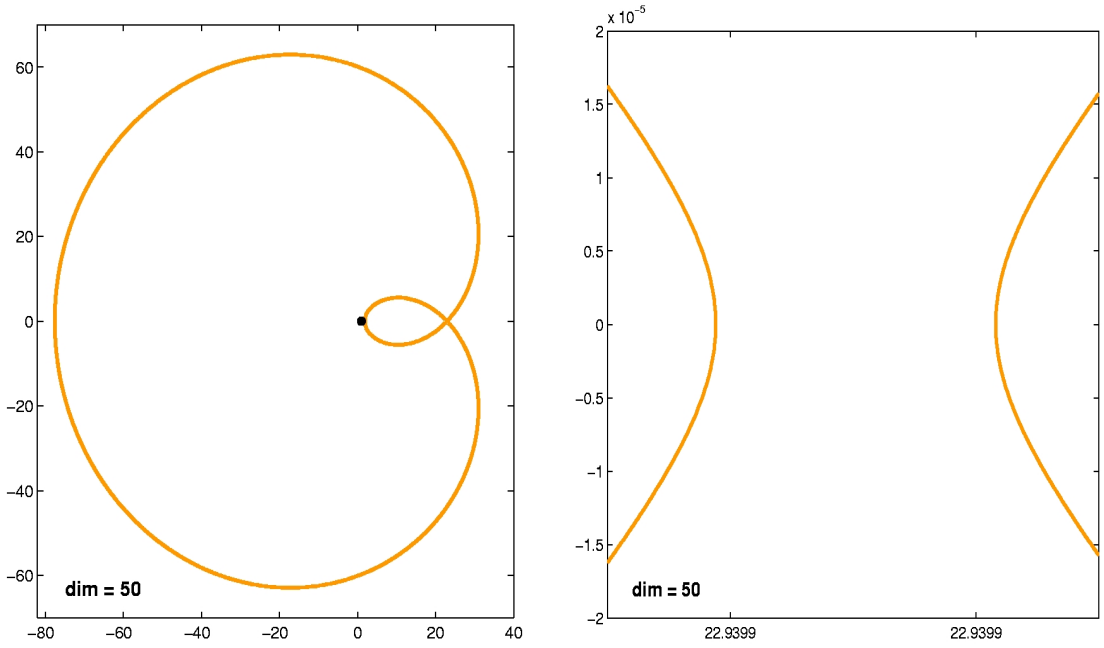

Figure 3.6: The ϵ-pseudospectrum of the 50×50 Demmel example shifted by $2I$ and $\epsilon = 10^{-0.9537}$ consist of the points inside the outer closed curve excluding the points inside the inner curve. On the right a close-up view of the region where the inner curve is close to being tangent to the outer curve is shown.

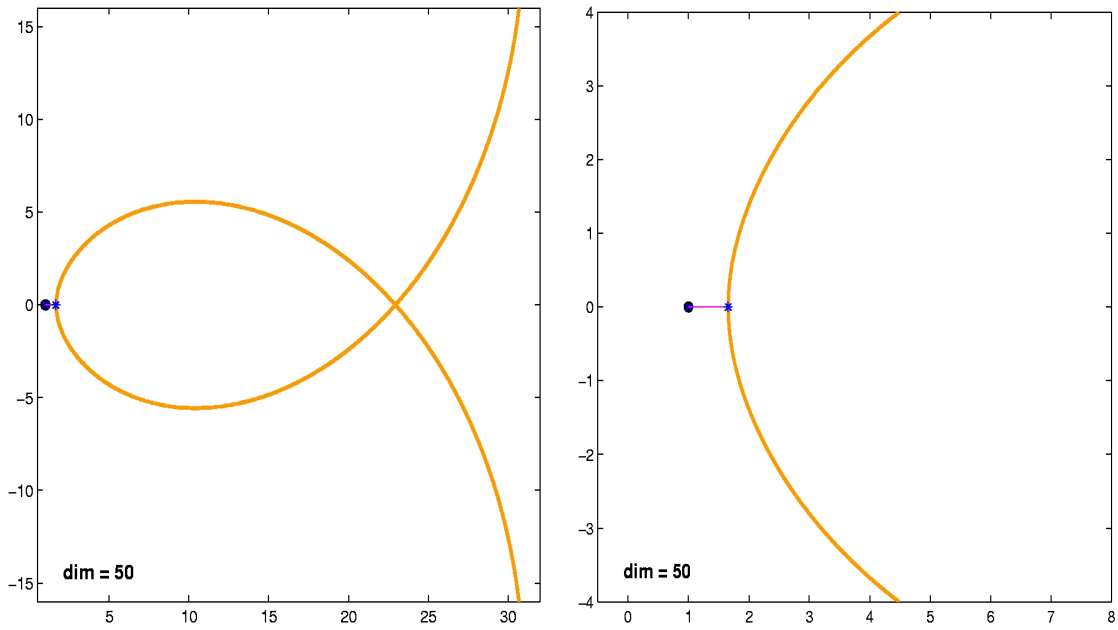

Figure 3.7: The radial search with $\theta = 0$ on the Demmel example fails to return an accurate result because of rounding errors. On the right a close-up view of the leftmost intersection is shown. The result of the radial search in finite precision is the leftmost intersection point marked with a blue asterisk.

only one point, approximately $r = 1.6571$, close to what we retrieved in the presence of rounding errors.

Notice that the radial search is a generalization of the horizontal search used in the computation of the ϵ-pseudospectral abscissa. Indeed the radial search above is the same as a horizontal search, since the search is performed in the direction $\theta = 0$. Even though the dependence of $\eta_\epsilon(\theta)$ on ϵ is discontinuous, the ϵ-pseudospectral radius is computed accurately as it is a continuous function of ϵ and, as we argued in §3.1.6, the algorithm is backward stable.

3.4.3 Running times

We tested the algorithms presented in this chapter for Grcar matrices of various sizes available through *EigTool* with $\epsilon = 0.1$. The running times of Algorithm 3, Algorithm 4 and the Boyd-Balakrishnan algorithm for the discrete distance to instability are listed in Table 3.1, Table 3.2 and Table 3.3, respectively. The input Grcar matrices to the algorithm for the discrete distance to instability are multiplied by 0.4 to ensure that all of their eigenvalues lie inside the unit circle.

Size	Total running time in secs	Running time per iteration in secs.
10	0.090	0.018
20	0.190	0.038
40	0.900	0.150
80	6.450	1.075
160	52.440	10.488
320	575.270	115.054

Table 3.1: The total and average running times in seconds per iteration of Algorithm 3 on Grcar examples of various size and $\epsilon = 10^{-1}$.

Size	Total running time in secs	Running time per iteration in secs.
10	0.070	0.012
20	0.120	0.024
40	0.490	0.082
80	3.160	0.527
160	40.660	6.777
320	406.580	81.316

Table 3.2: The total and average running time per iteration of Algorithm 4 on Grcar examples of various size.

In the tables we again include both the total running times and the average running times per iteration in seconds. The algorithms for the numerical radius and the discrete distance to instability are slightly faster than the algorithm for the pseudospectral radius, but the ratio of the running times for the algorithm for the pseudospectral radius and the algorithm for the numerical radius (or the algorithm for the discrete distance to instability) can be bounded by a constant for all sizes. Note also that when we compare the running times of the algorithms for the pseudospectral radius and abscissa (see §2.3.3), we observe that the algorithm for the pseudospectral abscissa requires less time on matrices of same size. This is mainly due to fact that we have to solve generalized eigenvalue problems in the algorithm for the pseudospectral radius as opposed to standard eigenvalue problems for the pseudospectral abscissa.

3.4.4 Extensions to matrix polynomials

We apply the extension of Algorithm 3 to matrix polynomials in order to retrieve the ϵ-pseudospectral radius of Q defined by (2.23) with $\gamma = [1\ 1\ 1]$ for $\epsilon = 0.1$,

Size	Total running time in secs	Running time per iteration in secs.
10	0.070	0.009
20	0.180	0.023
40	0.560	0.080
80	3.270	0.467
160	18.430	6.143
320	147.440	73.720

Table 3.3: The total and average running time per iteration of the Boyd-Balakrishnan algorithm in §3.3 for the discrete distance to instability on Grcar examples of various size.

0.3, 0.5, 0.7, 0.9 and with $\gamma = [0.1\ 1\ 0.1]$ for $\epsilon = 1, 3, 5, 7, 9$. For each γ and ϵ the point where the ϵ-pseudospectral radius is attained is marked with a black circle in Figure 3.8.

The quadratic matrix polynomial $\hat{Q}(\lambda) = \sum_{j=0}^{2} \lambda^j \hat{Q}_j$ with

$$\hat{Q}_2 = \begin{bmatrix} -27 & -81 & -162 & -162 \\ 6.75 & 0 & 0 & 0 \\ 0 & 6.75 & 0 & 0 \\ 0 & 0 & 6.75 & 0 \end{bmatrix}, \quad \hat{Q}_1 = \begin{bmatrix} 6 & 4.5 & 3 & 1.5 \\ 4.5 & 4.5 & 3 & 1.5 \\ 0 & 3 & 3 & 1.5 \\ 0 & 0 & 1.5 & 1.5 \end{bmatrix} \text{ and}$$
$$\hat{Q}_0 = \begin{bmatrix} -i & -0.5i & -1/3i & -0.25i \\ \pi & -i & -1/3i & -1/3i \\ i & \pi & -i & -0.5i \\ 0.5i & i & \pi & -i \end{bmatrix} \tag{3.38}$$

has all of its eigenvalues inside the unit circle. We compute its discrete distance instability with the scalings $\gamma = [1\ 1\ 1]$ and $\gamma = [0.1\ 1\ 0.1]$ using the Boyd-Balakrishnan algorithm which returns us the values 0.368 and 0.631, respectively. The ratio of the computed values 0.631/0.368 is equal to the ratio of the norms of the scaling vectors $\|[1\ 1\ 1]\|/\|[0.1\ 1\ 0.1]\| = \sqrt{3/1.02}$. (See the discussion at the end of §3.3.) In Figure 3.9 we also provide the plots of the function $h(\theta) = \sigma_{\min}[P(e^{i\theta})]/p_\gamma(1)$ for both of the scalings. Asterisks are used to mark the global minimizers of the functions.

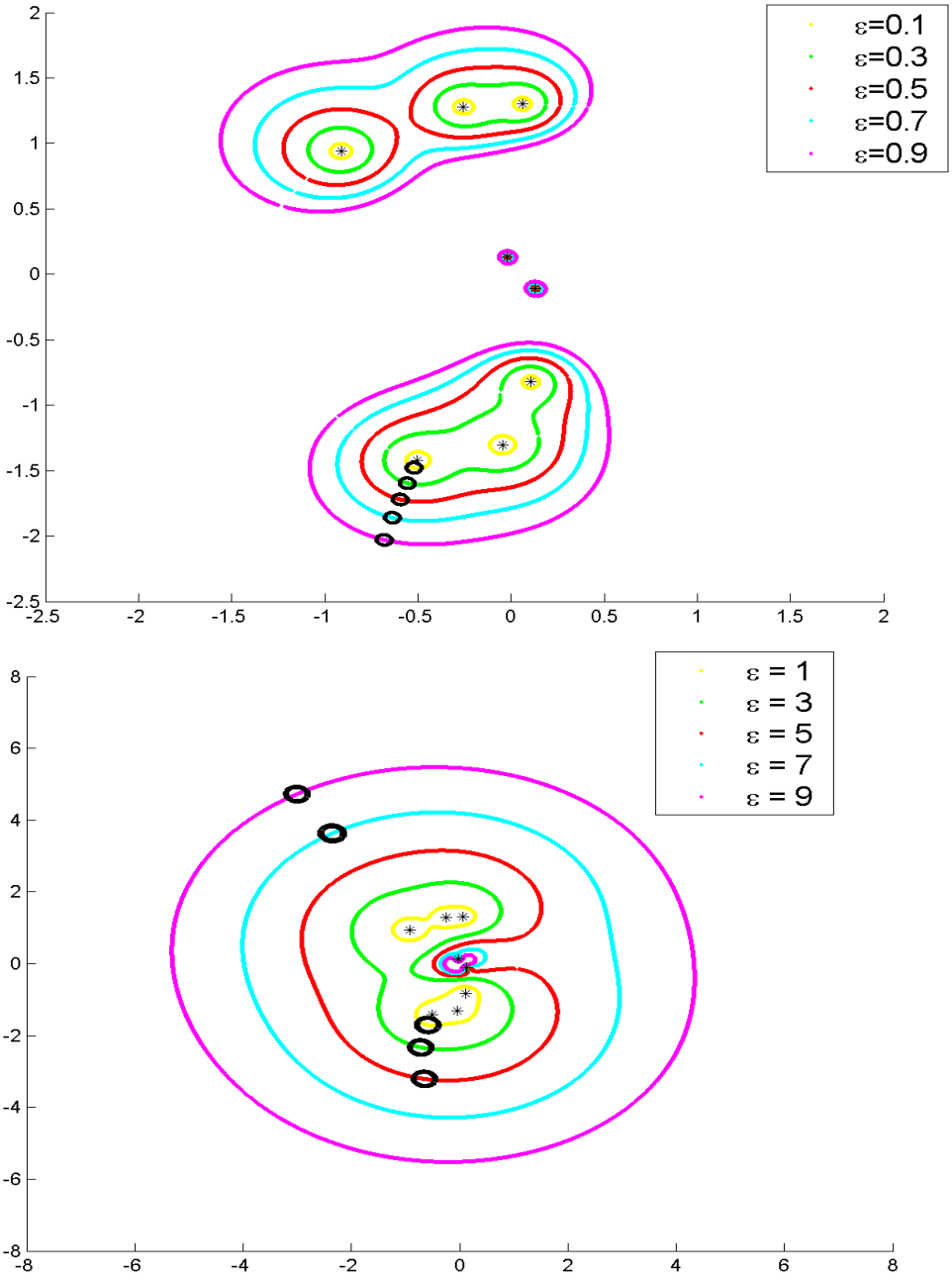

Figure 3.8: The points where the ϵ-pseudospectral radii are attained for the quadratic matrix polynomial Q defined by (2.23) are marked with black circles on the ϵ-pseudospectra with the scaling $\gamma = [1\ 1\ 1]$ at the top and $\gamma = [0.1\ 1\ 0.1]$ at the bottom. The asterisks indicate the location of the eigenvalues.

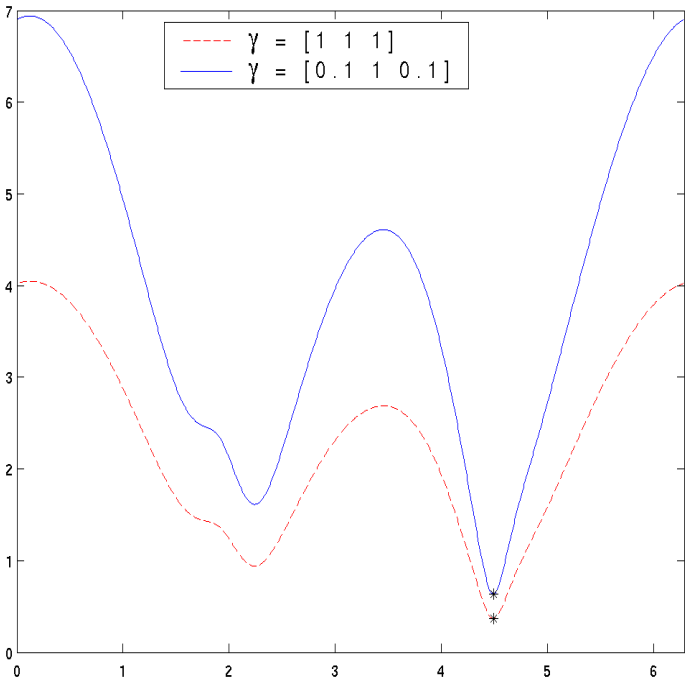

Figure 3.9: Graph of the function $h(\theta) = \sigma_{\min}(P(e^{i\theta}))/p_\gamma(1)$ for the quadratic matrix polynomial $\hat{Q}$ with $\gamma = [1\ 1\ 1]$ (dashed red curve) and $\gamma = [0.1\ 1\ 0.1]$ (solid blue curve) over the interval $[0, 2\pi)$.

Chapter 4

Distance to Uncontrollability for First-Order Systems

For the first-order system

$$x'(t) = Ax(t) + Bu(t), \tag{4.1}$$

the distance to uncontrollability defined by (1.20) has been shown to be equivalent to the singular value minimization problem

$$\tau(A, B) = \inf_{\lambda \in \mathbb{C}} \sigma_{\min}[A - \lambda I \quad B] \tag{4.2}$$

by Eising [24, 25]. Above and throughout this thesis for a rectangular matrix $X \in \mathbb{C}^{n \times n+m}$, $\sigma_{\min}(X)$ denotes the nth largest singular value of X. What makes the problem (4.2) considerably more challenging than (1.13), which is used for the computation of the distance to instability, is the necessity to optimize over the whole complex plane instead of over the imaginary axis or unit circle.

In this chapter we present an algorithm for low-precision approximation of the distance to uncontrollability in §4.2 and another one for high-precision approximation in §4.3. Both of the algorithms exploit the characterization (4.2). The algorithm for low-precision approximation is a progressive algorithm working on a grid. To compute an interval of length *tol* containing the distance to uncontrollability, it requires $O(\frac{n^3}{tol})$ operations. The algorithm for high-precision approximation modifies the algorithms by Gu [31] and Burke, Lewis and Overton [13] to reduce the overall complexity to $O(n^4)$ on average from $O(n^6)$. The effectiveness and reliability of the new methods are demonstrated by the numerical examples in §4.4. To facilitate the presentation of the algorithms in the next section, we first review the bisection idea for the distance to uncontrollability due to Gu and the trisection variant of Burke, Lewis and Overton, which allows approximation of the distance to uncontrollability for arbitrary precision.

4.1 Bisection and trisection

In [31] Gu introduced a bisection algorithm built upon the capability to verify one of the inequalities

$$\tau(A, B) \leq \delta_1 \tag{4.3}$$

and

$$\tau(A, B) > \delta_2 \tag{4.4}$$

for given $\delta_1 > \delta_2$. Notice that both of the inequalities may be satisfied in which case Gu's scheme in [31] returns information about only one of the inequalities. Gu's bisection algorithm (Algorithm 5) keeps only an upper bound on the distance to uncontrollability. It refines the upper bound until condition (4.4) is satisfied. At termination the distance to uncontrollability lies within a factor of 2 of δ_1, with $\delta_1/2 < \tau(A, B) \leq 2\delta_1$.

Algorithm 5 Gu's bisection algorithm for the distance to uncontrollability

Call:	$\delta_1 \leftarrow$ `Bisection`(A,B).
Input:	$A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$ with $m \leq n$.
Output:	A scalar δ_1 satisfying $\delta_1/2 < \tau(A, B) \leq 2\delta_1$.

```
Initialize the estimate as δ1 ← σmin([A  B])/2.
repeat
  δ2 ← δ1/2.
  Check which one of (4.3) and (4.4) holds.
  if (4.3) is verified then
    δ1 ← δ2.
    done ← FALSE.
  else
    % Otherwise (4.4) is verified.
    done ← TRUE.
  end if
until done = TRUE
Return δ1.
```

To obtain the distance to uncontrollability with better accuracy, Burke, Lewis and Overton [13] proposed a trisection variant. The trisection algorithm (Algorithm 6) bounds $\tau(A, B)$ by an interval $[L, U]$ and reduces the length of this interval by a factor of $\frac{2}{3}$ at each iteration (see Figure 4.1). Thus it can compute $\tau(A, B)$ to any desired accuracy.

What is crucial for both of the algorithms is the verification of (4.3) or (4.4). Our first strategy for the verification described in §4.2 is based on a grid and

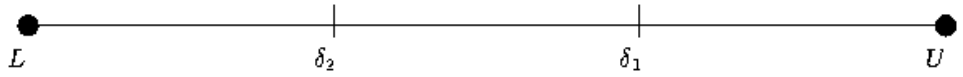

Figure 4.1: The trisection algorithm keeps track of an interval $[L, U]$ containing $\tau(A, B)$. At each iteration either L is updated to δ_2 or U is updated to δ_1.

Algorithm 6 Trisection algorithm of Burke, Lewis and Overton for the distance to uncontrollability

Call:	$[L,\ U] \leftarrow$ `Trisection`(A,B,ϵ).
Input:	$A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$ with $m \leq n$, and a tolerance $\epsilon > 0$.
Output:	Interval $[L, U]$ satisfying $L < \tau(A, B) \leq U$ and $U - L < \epsilon$.

Initialize the lower bound as $L \leftarrow 0$ and the upper bound as $U \leftarrow \sigma_{\min}([A\ \ B])$.
repeat
  $\delta_1 \leftarrow L + \frac{2}{3}(U - L)$
  $\delta_2 \leftarrow U + \frac{1}{3}(U - L)$
  Check which one of (4.3) and (4.4) holds.
  if (4.3) is verified **then**
    $U \leftarrow \delta_1$.
  else
    % otherwise (4.4) is verified.
    $L \leftarrow \delta_2$.
  end if
until $U - L < \epsilon$
Return $[L, U]$.

appropriate for low-precision estimation. The second strategy in §4.3 was also discussed in [32] and improves Gu's verification scheme used in [31] and [13], which requires the solution of eigenvalue problems of size $O(n^2)$ at each iteration and hence has complexity $O(n^6)$. We replace these eigenvalue problems with ones that can be efficiently solved by means of the real eigenvalue extraction technique that we introduce in §4.3.3, reducing the overall complexity to $O(n^4)$ on average and $O(n^5)$ in the worst case.

4.2 Low-precision approximation of the distance to uncontrollability

The idea of manipulating a grid for the computation of the distance to uncontrollability originates with Byers [16]. Using the fact that singular values are globally Lipschitz with Lipschitz constant one (from Weyl's Theorem, [38, Theorem 4.3.1]), it is straightforward to deduce

$$|\sigma_{\min}[A - \lambda_1 I \;\; B] - \sigma_{\min}[A - \lambda_2 I \;\; B]| \leq |\lambda_1 - \lambda_2| \tag{4.5}$$

for any complex pair λ_1, λ_2, that is the change in the minimum singular value is less than the norm of the perturbation $(\lambda_1 - \lambda_2)I$. This brings the idea of computing $\sigma_{\min}[A - \lambda I \; B]$ at various λ on a 2-D grid in the complex plane. If the distance between two adjacent grid points is h, the point in the complex plane where $\tau(A, B)$ is attained is at most $\frac{h}{\sqrt{2}}$ away from one of the grid points. Therefore $\tau(A, B)$ cannot differ from the minimum of the minimum singular values over the grid points by more than $\frac{h}{\sqrt{2}}$.

Indeed as the one-variable optimization problem

$$s(\alpha) = \inf_{\beta \in \mathbb{R}} \sigma_{\min}[A - (\alpha + \beta i)I \;\; B],$$

can be solved efficiently using a variant of the Boyd-Balakrishnan algorithm [9] for the distance to instability, we may work on a 1-D grid instead of a 2-D grid. Let α_* denote the real part of a point in the complex plane where $\tau(A, B)$ is attained, *i.e.*,

$$\tau(A, B) = \sigma_{\min}[A - (\alpha_* + \beta_* i)I \;\; B] \tag{4.6}$$

for some $\beta_* \in \mathbb{R}$ and let us assume *a priori* knowledge of a ν satisfying

$$\nu \geq |\alpha_*|. \tag{4.7}$$

A well known bound is $\nu = 2(\|A\| + \|B\|)$, but it may be possible to come up with tighter bounds for special cases. For any positive real h it follows from (4.5) that

$$|\tau(A, B) - \inf_{\nu \geq |jh|, j \in \mathbb{Z}} s(hj)| \leq h/2,$$

where the optimization of s is performed over a 1-D grid with two consecutive points differing by h. The inequality above must hold because the point where $\tau(A, B)$ is attained is at a distance less than or equal to $h/2$ from one of the vertical lines $\alpha = jh,\ \nu \geq |jh|$. These ideas are further elaborated in [16, 26, 33]. The trisection algorithm we present next benefits from a 1-D grid to verify either the upper bound (4.3) or the lower bound (4.4) at each iteration. The grid becomes finer as the algorithm focuses on the δ-level set of the function (δ changes from iteration to iteration)

$$g(\lambda) = \sigma_{\min}[A - \lambda I \ B]$$

and checks for points in this set whose real parts differ by smaller quantities at the later iterations.

We define the vertical cross section at α of the δ-level sets of $g(\lambda)$ as

$$\mathcal{S}_\delta(\alpha) = \{\beta : g(\alpha + \beta i) = \delta\}. \tag{4.8}$$

The next theorem states that for all $\alpha \in [\alpha_* - (\delta - \tau(A, B)), \alpha_* + (\delta - \tau(A, B))]$ the set $\mathcal{S}_\delta(\alpha)$ is nonempty.

Theorem 23. *Define α_* by (4.6) and assume $\delta > \tau(A, B)$ is given. For any $\alpha \in [\alpha_* - (\delta - \tau(A, B)), \alpha_* + (\delta - \tau(A, B))]$ there exists a real number β_α such that the equality*

$$\sigma_{\min}[A - (\alpha + \beta_\alpha i)I] = \delta$$

holds.

Proof. The function $\sigma_{\min}[A - (\alpha' + \beta_* i)I \ \ B]$ approaches ∞ as $\alpha' \to \infty$, where β_* is defined by (4.6). By the continuity of $\sigma_{\min}$ as a function of α', there exists a positive μ' such that

$$\sigma_{\min}[A - (\alpha_* + \mu' + \beta^* i)I \ \ B] = \delta.$$

Let μ_1 be the smallest positive μ' satisfying the equation above. Similarly, let μ_2 be the smallest positive μ' satisfying $\sigma_{\min}[A - (\alpha_* - \mu' + \beta_* i)I \ \ B] = \delta$. Note that for all $\alpha' \in [\alpha_* - \mu_2, \alpha_* + \mu_1]$, the inequality

$$\sigma_{\min}[A - (\alpha' + \beta_* i)I \ \ B] \leq \delta \tag{4.9}$$

holds. Furthermore from (4.5) we deduce $\mu_1 \geq \delta - \tau(A, B)$ and $\mu_2 \geq \delta - \tau(A, B)$.

Now choose any α such that $\alpha_* - (\delta - \tau(A, B)) \leq \alpha \leq \alpha_* + (\delta - \tau(A, B))$. Since α lies in the interval $[\alpha_* - \mu_2, \alpha_* + \mu_1]$ it follows from (4.9) that

$$\sigma_{\min}[A - (\alpha + \beta_* i)I \ \ B] \leq \delta. \tag{4.10}$$

As $\lim_{\beta\to\infty} \sigma_{\min}[A-(\alpha+\beta i)I \;\; B] = \infty$, the continuity of the minimum singular value as a function of β together with (4.10) imply that for some $\beta_\alpha \geq \beta_*$

$$\sigma_{\min}[A-(\alpha+\beta_\alpha i)I \;\; B] = \delta,$$

as desired. □

Whether the set $\mathcal{S}_\delta(\alpha)$ is empty or not can be verified by solving the Hamiltonian eigenvalue problem

$$D(\alpha,\delta) = \begin{bmatrix} -(A^*-\alpha I) & \delta I \\ \frac{BB^*}{\delta} - \delta I & A-\alpha I \end{bmatrix}. \tag{4.11}$$

We call this verification *the vertical search* at α. The next theorem, first proved in [16], relates the eigenvalues of $D(\alpha,\delta)$ and the points in the set $\mathcal{S}_\delta(\alpha)$.

Theorem 24. *Given a real α and a real $\delta \neq 0$, one of the singular values of $[A-(\alpha+\beta i)I \; B]$ is equal to δ if and only if βi is an eigenvalue of $D(\alpha,\delta)$.*

Proof. The nonnegative scalar δ is a singular value of $[A-(\alpha+\beta i)I \; B]$ if and only if the equations

$$[A-(\alpha+\beta i)I \; B] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \delta u$$
$$\begin{bmatrix} A^*-(\alpha-\beta i)I \\ B^* \end{bmatrix} u = \delta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

are satisfied simultaneously by some unit $u \in \mathbb{C}^n$ and $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \in \mathbb{C}^{n+m}$. From the bottom block of the second equation we have $v_2 = \frac{B^*}{\delta}u$. By plugging $\frac{B^*}{\delta}u$ into v_2 in the first equation and combining it with the upper block of the second equation, we obtain

$$\begin{bmatrix} -A^*+(\alpha-\beta i)I & \delta I \\ BB^*/\delta - \delta I & A-(\alpha+\beta i)I \end{bmatrix} \begin{bmatrix} u \\ v_1 \end{bmatrix} = 0,$$

which means that $i\beta$ is an eigenvalue of $D(\alpha,\delta)$ as desired. □

If $\beta i \in \Lambda(D(\alpha,\delta))$, then the theorem above implies $g(\alpha+\beta i) \leq \delta$ and we deduce from the intermediate value theorem that for some $\beta' \geq \beta$ the equality $g(\alpha+\beta' i) = \delta$ holds, so the vertical search at α succeeds. Otherwise, if the matrix $D(\alpha,\delta)$ does not have any imaginary eigenvalues, $g(\alpha+\beta i) > \delta$ for all β, meaning that the vertical search fails.

Putting together all the tools presented, we come up with a verification scheme to deduce one of the inequalities (4.3) and (4.4). Let $\delta = \delta_1$ and $\eta = 2(\delta_1 - \delta_2)$. We apply the vertical search at $-\nu, -\nu + \eta, \ldots, -\nu + \lceil \frac{2\nu}{\eta} \rceil \eta$. (Recall that ν is an upper bound on α_* in absolute value.) If any of the vertical searches returns an intersection point, then using definition (4.2) we can deduce the upper bound

$$\delta_1 = \delta \geq \tau(A, B).$$

If none of the vertical searches returns an intersection point, then suppose that the closest vertical line among $\alpha = -\nu + j\eta,\ j = 0, \ldots, \lceil \frac{2\nu}{\eta} \rceil$ to α_* is $\alpha = \alpha'$. Clearly

$$|\alpha' - \alpha_*| > \delta - \tau(A, B) \tag{4.12}$$

because otherwise according to Theorem 23 the set $\mathcal{S}_\delta(\alpha')$ would not be empty as verified. Furthermore, since α' is the closest vertical line to α_*, we have

$$|\alpha' - \alpha_*| \leq \eta/2. \tag{4.13}$$

Combining the inequalities (4.12) and (4.13) gives us

$$\delta - \tau(A, B) < \eta/2$$

or equivalently

$$\delta_2 < \tau(A, B).$$

Therefore Algorithm 6 is applicable. Given an interval $[L, U]$ containing the distance to uncontrollability, we set $\delta_1 = 2(U - L)/3$ and $\delta_2 = (U - L)/3$ and apply the verification scheme described. If any of the vertical searches succeeds we can update the upper bound U to δ_1. Otherwise, we refine the lower bound L to δ_2. So in either case, the interval length $U - L$ is reduced by a factor of two-thirds.

Clearly each iteration costs $O(\frac{\nu}{\eta} n^3)$, since we perform the vertical search at $\lceil \frac{2\nu}{\eta} \rceil + 1$ different positions. For higher precision we need to set η smaller, and therefore the later iterations are more expensive.

4.3 High-precision approximation of the distance to uncontrollability

Unlike the methods in the previous section, Gu's bisection method [31] and its trisection variant by Burke, Lewis and Overton [13] are not based on a grid. The computational cost of each iteration is fixed and only depends on the size of the input matrix for both of the algorithms. At each iteration for the verification

of one of (4.3) and (4.4) they require the extraction of the real eigenvalues of a pencil of size $2n^2 \times 2n^2$ and the imaginary eigenvalues of matrices of size $2n \times 2n$. Computationally the verification scheme is dominated by the extraction of the real eigenvalues of the pencil of size $2n^2 \times 2n^2$, which requires $O(n^6)$ operations if the standard QZ algorithm is used [30, Section 7.7].

In this section we present an alternative verification scheme for comparisons (4.3) and (4.4). In this new verification scheme we still need to find real eigenvalues of $2n^2 \times 2n^2$ matrices, so there is no asymptotic gain over Gu's verification scheme when we use the QR algorithm. Nevertheless, we show that the inverse of these $2n^2 \times 2n^2$ matrices shifted by a real number times the identity can be multiplied onto a vector efficiently by solving a Sylvester equation of size $2n$ with a cost of $O(n^3)$. Therefore, given a real number as the shift, by applying shifted inverse iteration or a shift-and-invert preconditioned Arnoldi method the closest eigenvalue to the real number can be obtained by performing $O(n^3)$ operations. Motivated by the fact that we only need real eigenvalues, we provide a divide-and-conquer type algorithm that scans the real axis to find the desired eigenvalues. The approach requires an upper bound on the norm of the input matrix (of size $2n^2 \times 2n^2$) as a parameter. Choosing this parameter arbitrarily large does not affect the efficiency of the algorithm much, though it may cause accuracy problems. We prove that extracting all of the real eigenvalues with the divide-and-conquer approach requires $O(n^4)$ operations on average and $O(n^5)$ operations in the worst case.

In §4.3.1 we review Gu's scheme for verifying which one of (4.3) and (4.4) holds. In §4.3.2 we present our modified eigenvalue problem for the same purpose and show how the closest eigenvalue to a given point for the modified problem can be computed efficiently. We introduce the divide-and-conquer approach for real eigenvalue extraction based on the closest eigenvalue computations to various points on the real axis in §4.3.3. In §4.3.4 we discuss some details related to the computation of the distance to uncontrollability using the real eigenvalue extraction technique, including methods to solve Sylvester equations and accuracy issues.

4.3.1 Gu's verification scheme

In [31] and [13] the determination of which one of the inequalities (4.3) and (4.4) holds is based on the following theorem [31], which is a consequence of (4.5).

Theorem 25 (Gu [31]). *Assume that $\delta > \tau(A, B)$ is given. Given an $\eta \in [0, 2(\delta - \tau(A, B))]$, there exist at least two pairs of real numbers α and β such that*

$$\delta \in \sigma\left([A - (\alpha + \beta i)I, B]\right) \text{ and } \delta \in \sigma\left([A - (\alpha + \eta + \beta i)I, B]\right), \tag{4.14}$$

where $\sigma(\cdot)$ denotes the set of singular values of its argument.

Theorem 23, on which the verification scheme of the previous section is based, is inspired by Theorem 25, but neither of the theorems is a generalization of the other. In Theorem 23 we proved the existence of an interval of length $2(\delta-\tau(A,B))$ containing α_* (the real part of the point where $\tau(A,B)$ is attained) exactly in the middle, such that for all α' in the interval, the set of intersection points of the δ-level set of $g(\lambda) = \sigma_{\min}[A - \lambda I \;\; B]$ and the vertical line $\alpha = \alpha'$, $\mathcal{S}_\delta(\alpha')$ is nonempty. On the other hand Theorem 25 states that for all $\eta \leq 2(\delta - \tau(A,B))$ there exist two distinct horizontal lines whose intersections with the δ-sublevel set of $g(\lambda)$ contain line segments of length η.

Suppose we set $\delta_1 = \delta$ and $\delta_2 = \delta - \eta/2$. The theorem above implies that, when no pair satisfying (4.14) exists, the inequality $\eta > 2(\delta - \tau(A,B))$ is satisfied, so condition (4.4) holds. On the other hand, when a pair exists, then by definition (4.2) we can conclude (4.3). Therefore if we have a procedure to verify the existence of a pair α and β satisfying (4.14) for a given δ and η, Algorithm 6 can be used to retrieve the distance to uncontrollability to an arbitrary precision.

By means of Gu's test which we describe next we can numerically verify whether a real pair of solutions to (4.14) exists. Equation (4.14) in Theorem 25 implies that there exist non-zero vectors $\begin{bmatrix} x \\ y \end{bmatrix}$, z, $\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$, and $\hat{z}$ such that

$$[A - (\alpha + \beta i)I \; B] \begin{bmatrix} x \\ y \end{bmatrix} = \delta z, \; \begin{bmatrix} A^* - (\alpha - \beta i)I \\ B^* \end{bmatrix} z = \delta \begin{bmatrix} x \\ y \end{bmatrix}, \quad (4.15a)$$

$$[A - (\alpha + \eta + \beta i)I \; B] \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \delta \hat{z}, \; \begin{bmatrix} A^* - (\alpha + \eta - \beta i)I \\ B^* \end{bmatrix} \hat{z} = \delta \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}. \quad (4.15b)$$

These equations can be rewritten as

$$\begin{bmatrix} -\delta I & A - \alpha I & B \\ A^* - \alpha I & -\delta I & 0 \\ B^* & 0 & -\delta I \end{bmatrix} \begin{bmatrix} z \\ x \\ y \end{bmatrix} = \beta i \begin{bmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ x \\ y \end{bmatrix} \quad (4.16a)$$

and

$$\begin{bmatrix} -\delta I & A - (\alpha + \eta)I & B \\ A^* - (\alpha + \eta)I & -\delta I & 0 \\ B^* & 0 & -\delta I \end{bmatrix} \begin{bmatrix} \hat{z} \\ \hat{x} \\ \hat{y} \end{bmatrix} = \beta i \begin{bmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{z} \\ \hat{x} \\ \hat{y} \end{bmatrix}. \quad (4.16b)$$

Furthermore using the QR factorization

$$\begin{bmatrix} B \\ -\delta I \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (4.17)$$

these problems can be reduced to standard eigenvalue problems of size $2n \times 2n$, *i.e.* the eigenvalues of the pencils in (4.16a) and in (4.16b) are the same as the eigenvalues of the matrices

$$\begin{bmatrix} A - \alpha I & BQ_{22} - \delta Q_{12} \\ \delta Q_{12}^{-1} & -Q_{12}^{-1}(A^* - \alpha I)Q_{12} \end{bmatrix} \tag{4.18a}$$

and

$$\begin{bmatrix} A - (\alpha + \eta)I & BQ_{22} - \delta Q_{12} \\ \delta Q_{12}^{-1} & -Q_{12}^{-1}(A^* - (\alpha + \eta)I)Q_{12} \end{bmatrix} \tag{4.18b}$$

respectively. In order for (4.14) to have at least one real solution (α, β), these two matrices must share a common imaginary eigenvalue βi. This requires a $2n^2 \times 2n^2$ generalized eigenvalue problem to have a real eigenvalue α (see [31]). For a given δ and η, we check whether the latter generalized eigenvalue problem has any real eigenvalue α. If it does, then we check the existence of a real eigenvalue α for which the matrices (4.18a) and (4.18b) share a common pure imaginary eigenvalue βi. There exists a pair of α and β satisfying (4.14) if and only if this process succeeds.

4.3.2 Modified fast verification scheme

It turns out that Gu's verification scheme can be simplified. In this modified scheme the $2n^2 \times 2n^2$ generalized eigenvalue problems whose real eigenvalues are sought in Gu's scheme are replaced by $2n^2 \times 2n^2$ standard eigenvalue problems, and the $2n \times 2n$ matrices (4.18a) and (4.18b) whose imaginary eigenvalues are sought are replaced by the matrices $D(\alpha, \delta)$ and $D(\alpha + \eta, \delta)$ defined by (4.11) of size $2n \times 2n$.

The simplification of the problem of size $2n^2 \times 2n^2$ is significant, as the inverse of the new matrix of size $2n^2 \times 2n^2$ (whose real eigenvalues are sought) times a vector can be computed in a cheap manner by solving a Sylvester equation of size $2n \times 2n$ with a cost of $O(n^3)$. As a consequence the closest eigenvalue to a given complex point can be computed efficiently by applying shifted inverse iteration or shift-and-invert Arnoldi. We discuss how this idea can be extended to extract all of the real eigenvalues with an average cost of $O(n^4)$ and a worst-case cost of $O(n^5)$, reducing the running time of each iteration of the bisection or the trisection algorithm asymptotically.

New generalized eigenvalue problem

If there exists a pair (α, β) satisfying (4.14), the Hamiltonian matrices $D(\alpha, \delta)$ and $D(\alpha+\eta, \delta)$ must share the eigenvalue $i\beta$. The Hamiltonian property implies that the matrices $D(\alpha+\eta, \delta)$ and $-D(\alpha+\eta, \delta)^*$ have the same set of eigenvalues.

For $D(\alpha,\delta)$ and $D(\alpha+\eta,\delta)$ or equivalently $D(\alpha,\delta)$ and $-D(\alpha+\eta,\delta)^*$ to share a common eigenvalue βi, the matrix equation

$$D(\alpha,\delta)X + X(D(\alpha+\eta,\delta))^* = 0 \tag{4.19}$$

$$D(0,\delta)X + X(D(\eta,\delta))^* = \alpha \left(\begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} X + X \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \right) \tag{4.20}$$

must have a nonzero solution X [39, Theorem 4.4.6]. Let

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}.$$

Notice that in (4.20) all the terms depending on α are collected on the right-hand side. Therefore writing the matrix equation (4.20) in a vector form yields the generalized eigenvalue problem

$$(\mathcal{F}(\delta,\eta) - \alpha\mathcal{G}) \begin{bmatrix} \mathbf{vec}(X_{11}) \\ \mathbf{vec}(X_{12}) \\ \mathbf{vec}(X_{21}) \\ \mathbf{vec}(X_{22}) \end{bmatrix} = 0, \tag{4.21}$$

where

$$\mathcal{F}(\delta,\eta) = \begin{bmatrix} -A_1^* - A_2^T & \delta I & \delta I & 0 \\ B_2^T & -A_1^* + \bar{A}_2 & 0 & \delta I \\ B_1 & 0 & A_1 - A_2^T & \delta I \\ 0 & B_1 & B_2^T & A_1 + \bar{A}_2 \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} -2I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2I \end{bmatrix},$$

$\mathbf{vec}(X)$ denotes the vector formed by stacking the column vectors of X, $A_1 = I \otimes A$, $A_2 = (A - \eta I) \otimes I$, $B_1 = I \otimes \hat{B}$, $B_2 = \hat{B} \otimes I$ and $\bar{A}_2$ denotes the matrix obtained by taking the complex conjugate of A_2 entrywise. The block rows of (4.21) are obtained by equating the left-hand sides and right-hand sides in (4.20) block by block in vector form. The first, second, third and final block rows in (4.21) correspond to the upper left block, upper right block, lower left block and lower right block in (4.20) in vector form. For this derivation we used the property $\mathbf{vec}(AXB) = (B^T \otimes A)\mathbf{vec}(X)$.

Half of the eigenvalues of the pencil $\mathcal{F}(\delta,\eta) - \mathcal{G}$ are at infinity. Deflating the infinite eigenvalues, or equivalently eliminating the variables $\mathbf{vec}(X_{12})$ and $\mathbf{vec}(X_{21})$ in (4.21), leads us to the standard eigenvalue problem

$$\mathcal{A}v = \alpha v, \tag{4.22}$$

where

$$\mathcal{A} = \frac{1}{2}\left(\begin{bmatrix} A_1^* + A_2^T & 0 \\ 0 & A_1 + \bar{A}_2 \end{bmatrix} - \begin{bmatrix} -\delta I & -\delta I \\ B_1 & B_2^T \end{bmatrix} \begin{bmatrix} -A_1^* + \bar{A}_2 & 0 \\ 0 & A_1 - A_2^T \end{bmatrix}^{-1} \begin{bmatrix} B_2^T & \delta I \\ B_1 & \delta I \end{bmatrix} \right). \tag{4.23}$$

It should be noted that the deflation of infinite eigenvalues and the conversion of the generalized eigenvalue problem $\mathcal{F}(\delta, \eta) - \lambda \mathcal{G}$ into the standard eigenvalue problem (4.22) are possible under the assumption that $A_1 - A_2^T$ is nonsingular, or equivalently, A does not have two eigenvalues that differ by η. If A has two eigenvalues whose difference is close to η, the Kronecker product matrix $\mathcal{A}$ is ill-conditioned. We discuss accuracy issues at the end of this section.

For the verification of a pair α and β satisfying (4.14), we first solve the eigenvalue problem (4.22). If there exists a real eigenvalue α, the matrices $D(\alpha, \delta)$ and $D(\alpha + \eta, \delta)$ share an eigenvalue but not necessarily an imaginary one. Therefore at a second step for each real $\alpha \in \Lambda(\mathcal{A})$ we must check whether the common eigenvalue of $D(\alpha, \delta)$ and $D(\alpha+\eta, \delta)$ is imaginary. A pair satisfying (4.14) exists if and only if both of the steps succeed.

Inverse iteration

The eigenvalue problem in (4.22) is a simplified version of the generalized eigenvalue problem in [31]. This is a problem of finding the real eigenvalues of a nonsymmetric matrix. The implementation [14] of the trisection algorithm introduced in [13] uses the MATLAB function `eig` to compute the eigenvalues of that generalized eigenvalue problem with a cost of $O(n^6)$ and therefore does not exploit the fact that we only need the real eigenvalues of the generalized problem. In §4.3.3 we discuss a divide-and-conquer approach to extract the real eigenvalues of a given matrix $\mathcal{X}$ that is preferable to `eig` when the closest eigenvalue of $\mathcal{X}$ to a given point can be obtained efficiently.

In this section we show how one can compute the closest eigenvalue of $\mathcal{A}$ to a given point in the complex plane in $O(n^3)$ time. This is due to the fact that given a shift ν and a vector $u \in \mathbb{C}^{2n^2}$, the multiplication $(\mathcal{A} - \nu I)^{-1} u$ can be performed by solving a Sylvester equation of size $2n \times 2n$ which is derived next. Therefore, shifted inverse iteration or shift-and-invert Arnoldi can locate the closest eigenvalue efficiently.

We start with the simplified case $v = \mathcal{A}^{-1} u$, where $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$, $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ and $u_1, u_2, v_1, v_2 \in \mathbb{C}^{n^2}$. We essentially reverse the derivation of the eigenvalue problem (4.22). We need to solve the linear system

$$\mathcal{A} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}. \tag{4.24}$$

Making the substitution

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} -A_1^* + \bar{A}_2 & 0 \\ 0 & A_1 - A_2^T \end{bmatrix}^{-1} \begin{bmatrix} B_2^T & \delta I \\ B_1 & \delta I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{4.25}$$

in (4.24) yields

$$\begin{bmatrix} A_1^* + A_2^T & 0 \\ 0 & A_1 + \bar{A}_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} - \begin{bmatrix} -\delta I & -\delta I \\ B_1 & B_2^T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{4.26}$$

By combining (4.25) and (4.26) we obtain a linear system of double size,

$$\begin{bmatrix} A_1^* + A_2^T & \delta I & \delta I & 0 \\ B_2^T & A_1^* - \bar{A}_2 & 0 & \delta I \\ B_1 & 0 & -A_1 + A_2^T & \delta I \\ 0 & -B_1 & -B_2^T & A_1 + \bar{A}_2 \end{bmatrix} \begin{bmatrix} v_1 \\ w_1 \\ w_2 \\ v_2 \end{bmatrix} = 2 \begin{bmatrix} u_1 \\ 0 \\ 0 \\ u_2 \end{bmatrix}. \tag{4.27}$$

Notice that the matrix on the left-hand side of the equation above when the signs of A_1 and A_2 are negated and the sign of the bottom block row is negated is same as $\mathcal{F}(\delta, \eta)$. The matrix $\mathcal{F}(\delta, \eta)$ is the left-hand side of (4.20) in vectoral form. Therefore if we reverse the procedure by introducing

$$u = \begin{bmatrix} \mathbf{vec}(U_1) \\ \mathbf{vec}(U_2) \end{bmatrix}, \; v = \begin{bmatrix} \mathbf{vec}(V_1) \\ \mathbf{vec}(V_2) \end{bmatrix}, \; w = \begin{bmatrix} \mathbf{vec}(W_1) \\ \mathbf{vec}(W_2) \end{bmatrix},$$

the resulting matrix equation must be

$$\begin{bmatrix} A^* & \delta I \\ \hat{B} & -A \end{bmatrix} Z + Z \begin{bmatrix} A - \eta I & \hat{B} \\ \delta I & -A^* + \eta I \end{bmatrix} = 2 \begin{bmatrix} U_1 & 0 \\ 0 & -U_2 \end{bmatrix}, \tag{4.28}$$

that is the signs of A and $A - \eta I$ as well as the sign of the lower right block of the matrix on the left-hand side in (4.20) are negated, where

$$Z = \begin{bmatrix} V_1 & W_1 \\ W_2 & V_2 \end{bmatrix}. \tag{4.29}$$

The derivation easily extends to the multiplication $(\mathcal{A} - \nu I)^{-1} u$ for a given shift ν. Starting from the linear system

$$(\mathcal{A} - \nu I) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{4.30}$$

and applying the same steps, we end up with

$$\begin{bmatrix} A_1^* + A_2^T - 2\nu I & \delta I & \delta I & 0 \\ B_2^T & A_1^* - \bar{A}_2 & 0 & \delta I \\ B_1 & 0 & -A_1 + A_2^T & \delta I \\ 0 & -B_1 & -B_2^T & A_1 + \bar{A}_2 - 2\nu I \end{bmatrix} \begin{bmatrix} v_1 \\ w_1 \\ w_2 \\ v_2 \end{bmatrix} = 2 \begin{bmatrix} u_1 \\ 0 \\ 0 \\ u_2 \end{bmatrix}. \tag{4.31}$$

In terms of a matrix equation, we obtain

$$\begin{bmatrix} A^* - \nu I & \delta I \\ \hat{B} & -A + \nu I \end{bmatrix} Z + Z \begin{bmatrix} A - (\eta + \nu) I & \hat{B} \\ \delta I & -A^* + (\eta + \nu) I \end{bmatrix} = 2 \begin{bmatrix} U_1 & 0 \\ 0 & -U_2 \end{bmatrix} \tag{4.32}$$

where Z is as defined in (4.29). Equation (4.32) is identical to (4.28) except that A is replaced by $A - \nu I$ in (4.28). It is a Sylvester equation of size $2n \times 2n$. Therefore Z from which we can retrieve $v = (\mathcal{A} - \nu I)^{-1} u$ can be computed by using a Sylvester equation solver (such as the LAPACK routine `dtrsyl` [1]).

4.3.3 Divide-and-conquer algorithm for real eigenvalue extraction

In this section we seek the real eigenvalues of a given matrix $\mathcal{X} \in \mathbb{C}^{q \times q}$. The divide-and-conquer approach here is preferable to the standard ways of computing eigenvalues, such as the QR algorithm, when $(\mathcal{X} - \nu I)^{-1} u$ is efficiently computable for a given shift $\nu \in \mathbb{R}$ and a given vector $u \in \mathbb{C}^q$. In particular, as discussed in the previous section, this is the case for $\mathcal{A}$. In [32] a more brute force approach called adaptive progress was also suggested for real eigenvalue extraction as an alternative to the divide-and-conquer approach.

Throughout this section we will assume the existence of a reliable implementation of the shifted inverse iteration or a shift-and-invert Arnoldi method that returns the closest eigenvalue to a given shift accurately. In practice we make use of the MATLAB function `eigs` (based on ARPACK [48, 49]). Additionally, we assume that an upper bound, D, on the norm of $\mathcal{X}$ is available and therefore we know that all of the real eigenvalues lie in the interval $[-D, D]$. A straightforward approach would be to partition the interval $[-D, D]$ into equal subintervals and find the closest eigenvalue to the midpoint of each interval. This approach must work as long as the subintervals are chosen small enough. Nevertheless, partitioning $[-D, D]$ into very fine subintervals is not desirable, since this will require an excessive number of closest eigenvalue computations.

In the divide-and-conquer approach, given an interval $[L, U]$ we compute the eigenvalue of $\mathcal{X}$ closest to the midpoint of the interval $\nu = \frac{U+L}{2}$. If the modulus of the difference between the computed eigenvalue λ and the midpoint ν is greater than half of the length of the interval, then we terminate. Otherwise we apply the same procedure to the subintervals $[L, \nu - |\lambda - \nu|]$ and $[\nu + |\lambda - \nu|, U]$. Initially we apply the algorithm to the whole interval $[-D, D]$.

Figure 4.2 and 4.3 illustrates the first four iterations of the divide-and-conquer algorithm on an example. The divide-and-conquer algorithm completes the investigation of the real interval where the real eigenvalues reside after iterating 7 times.

Algorithm 7 Divide-and-conquer real eigenvalue search algorithm

Call:	$\Lambda \leftarrow$ `Divide_And_Conquer`$(\mathcal{X},L,U)$.
Input:	$\mathcal{X} \in \mathbb{C}^{q \times q}$, a lower bound L for the smallest real eigenvalue desired and an upper bound U for the largest real eigenvalue desired.
Output:	$\Lambda \in \mathbb{R}^l$ with $l \leq q$ containing all of the real eigenvalues of $\mathcal{X}$ in the interval $[L, U]$.

Set the shift $\nu \leftarrow \frac{(U+L)}{2}$.
Compute the eigenvalue λ closest to the shift ν.
if $U - L < 2|\lambda - \nu|$ **then**
  % Base case: there is no real eigenvalue in the interval $[L, U]$.
  Return [ ].
else
  % Recursive case: Search the left and right intervals.
  $\Lambda_L \leftarrow$ `Divide_And_Conquer`$(\mathcal{X},L,\nu - |\lambda - \nu|)$
  $\Lambda_R \leftarrow$ `Divide_And_Conquer`$(\mathcal{X},\nu + |\lambda - \nu|,U)$
  % Combine all of the real eigenvalues.
  if λ is real **then**
    Return $\lambda \cup \Lambda_L \cup \Lambda_R$.
  else
    Return $\Lambda_L \cup \Lambda_R$.
  end if
end if

For reliability the parameter D must be chosen large. Suppose that all the eigenvalues are contained in the disk of radius D' with $D' \ll D$. To discover that there is no real eigenvalue in the interval $[D', D]$ at most two extra closest eigenvalue computations are required. If the first shift tried in the interval $[D', D]$ is closer to D' rather than D, then the distance from the closest eigenvalue to this shift may be less than half of the length of the interval $[D', D]$, so a second closest eigenvalue computation may be needed. Otherwise the interval $[D', D]$ will be investigated in one iteration. Similar remarks hold for the interval $[-D, -D']$. However, the larger choices of D may slightly increase or decrease the number of shifts required to investigate $[-D', D']$. The important point is that regardless of how large D is compared to the radius of the smallest disk containing the eigenvalues, the cost is limited to approximately four extra iterations.

Next we show that the number of closest eigenvalue computations cannot exceed $2q + 1$ (recall that $\mathcal{X} \in \mathbb{C}^{q \times q}$).

Theorem 26 (**Worst Case for Algorithm 7**). *The number of closest eigenvalue computations made by Algorithm 7 is no more than* $2q + 1$.

Proof. We can represent the progress of the algorithm by a full binary tree, *i.e.* a tree with each node having either two children or no children. Each node of the tree corresponds to an interval. The root of the tree corresponds to the whole interval $[-D, D]$. At each iteration of the algorithm the interval under consideration is either completely investigated or replaced by two disjoint subintervals that need to be investigated. In the first case, the node corresponding to the current interval is a leaf. In the second case, the node has two children, one for each of the subintervals.

We claim that the number of leaves in this tree cannot exceed $q + 1$. The intervals corresponding to the leaves are disjoint. Each such interval has a closest left interval (except the leftmost interval) and a closest right interval (except the rightmost interval) represented by two of the leaves in the tree. Each interval is separated from the closest one on the left by the part of a disk on the real axis in which an eigenvalue lies and similarly for the closest interval on the right. Since the matrix $\mathcal{X}$ has q eigenvalues, there can be at most q separating disks and therefore there can be at most $q + 1$ disjoint intervals represented by the leaves of the tree. A full binary tree with $q + 1$ leaves has q internal nodes. Therefore, the total number of the nodes in the tree, which is the same as the number of closest eigenvalue computations, cannot exceed $2q + 1$. □

The upper bound $2q + 1$ on the worst-case performance of the algorithm is tight, as illustrated by the following example. Consider a matrix with the real eigenvalues $\frac{2^{j-1}-1}{2^{j-1}}$, $j = 1 \ldots q$, and suppose that we search over the interval

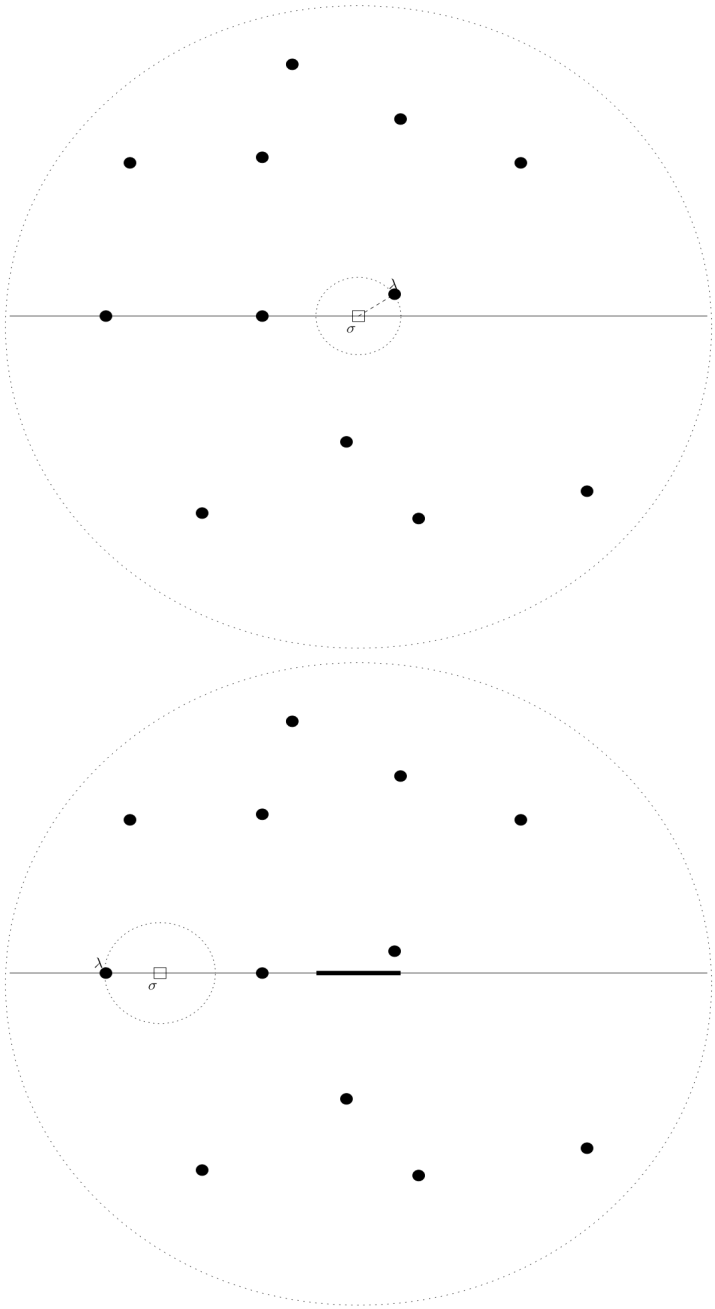

Figure 4.2: The first two iterations (the top one is the first iteration) of the divide-and-conquer algorithm on an example. Black dots denote the eigenvalues. Squares mark the location of the shift ν. The closest eigenvalue to ν is denoted by λ. The part of the real axis already investigated is marked by a thicker line.

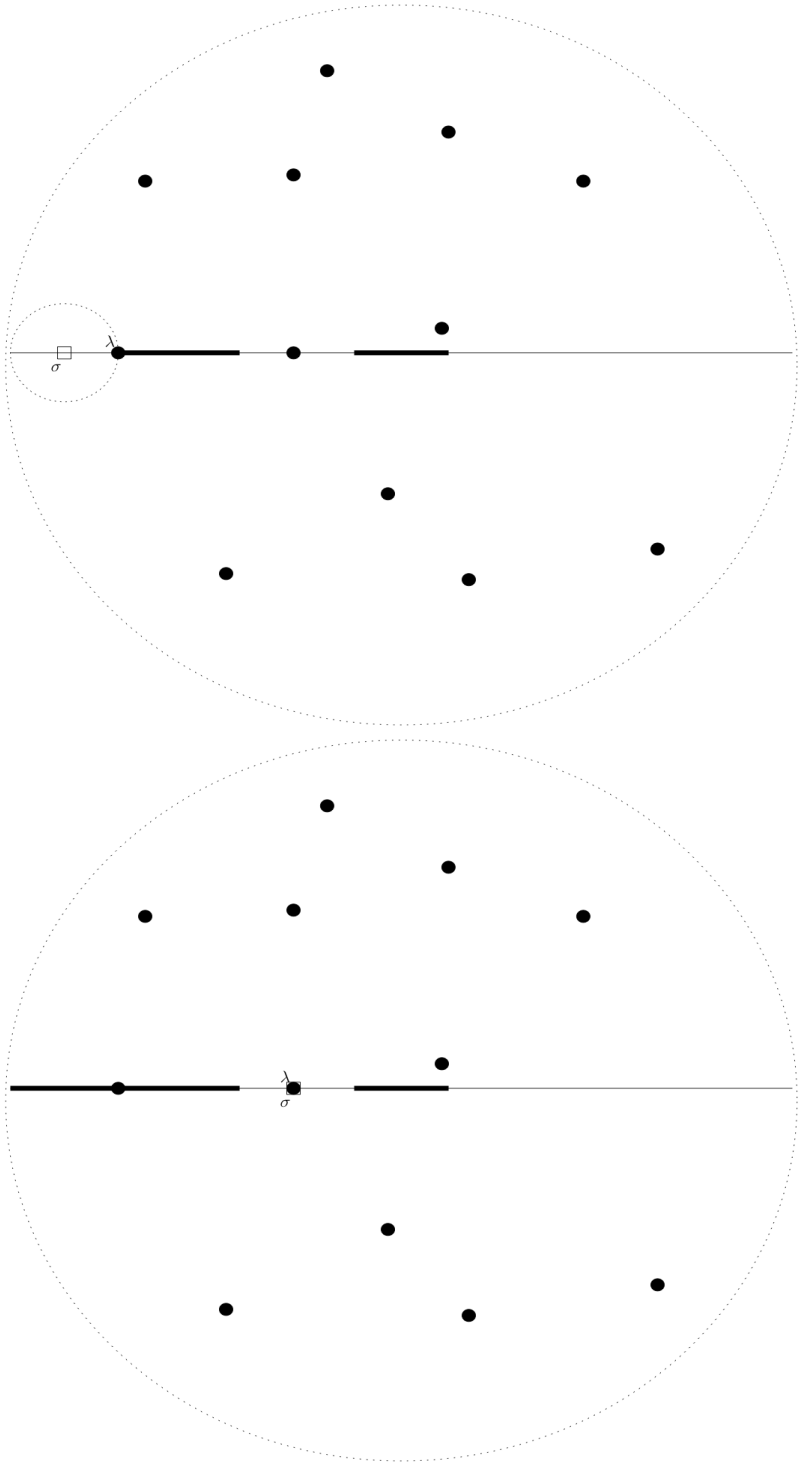

Figure 4.3: The third and fourth iterations of the divide-and-conquer algorithm on the example in Figure 4.2.

$[-1, 1]$. Clearly, the algorithm discovers each eigenvalue twice except the largest one, which it discovers three times (this is assuming that when there are two eigenvalues equally close to a midpoint, the algorithm locates the eigenvalue on the right). Therefore, the total number of closest eigenvalue computations is $2q + 1$.

Next we aim to show that the average-case performance of the algorithm is much better than the worst case. First we note the following elementary result that is an immediate consequence of the fact that the square-root function is strictly concave.

Lemma 27. *Given l positive distinct integers $k_1, k_2, \ldots, k_l$ and l real numbers $p_1, p_2, \ldots, p_l \in (0, 1)$ such that $\sum_{j=1}^{l} p_j = 1$, the inequality*

$$\sqrt{\sum_{j=1}^{l} p_j k_j} > \sum_{j=1}^{l} p_j \sqrt{k_j} \tag{4.33}$$

holds.

In the average-case analysis we let the eigenvalues of $\mathcal{X}$, say $\xi_1, \xi_2, \ldots \xi_q$, vary. We assume that the eigenvalues are independently selected from a uniform distribution inside the circle centered at the origin with radius μ. We use Algorithm 7 to compute the real eigenvalues lying inside the circle of radius $D = 1 \leq \mu$ (the value of the radius D is irrelevant for the average-case analysis as discussed below; we choose $D = 1$ for simplicity). In Table 4.1 the random variables and the probability density functions referenced by the proof of the next theorem are summarized.

The quantity we are interested in is $E(X|N = j)$, the expected number of iterations required by Algorithm 7 given that there are j eigenvalues inside the unit circle. We list a few observations.

- *The eigenvalues $\omega_1, \omega_2, \ldots, \omega_j$ contained in the circle of radius $D = 1$ are uniformly distributed and mutually independent:* This is a simple consequence of the assumption that the eigenvalues are selected from uniform distributions and mutually independently. Let the eigenvalues inside the unit circle be $\xi_{i_1}, \xi_{i_2}, \ldots \xi_{i_j}$ with $i_1 < i_2 < \cdots < i_j$. We associate ω_k with the location of the kth smallest indexed eigenvalue inside the unit circle, *i.e.* $\omega_k = \xi_{i_k}$. Let C_1 denote the unit circle. The variable ω_k is uniformly

distributed because

$$\begin{aligned} p(\omega_k|j \text{ of } \xi_1, \ldots \xi_q \in C_1) &= \textstyle\sum_{i_1,\ldots i_j} \quad (p(\xi_{i_1}, \ldots \xi_{i_j} \in C_1|j \text{ of } \xi_1, \ldots \xi_q \in C_1) \\ &\qquad\qquad p(\omega_k|\xi_{i_1}, \ldots \xi_{i_j} \in C_1)) \\ &= \textstyle\sum_{i_1,\ldots i_j} \quad \left(\begin{array}{c} q \\ j \end{array} \right)^{-1} p(\omega_k|\xi_{i_k} \in C_1) \\ &= \textstyle\sum_{i_1,\ldots i_j} \quad \left(\begin{array}{c} q \\ j \end{array} \right)^{-1} \frac{1}{\pi} = \frac{1}{\pi}. \end{aligned}$$

Above the summation is over the subsets of $\xi_1, \xi_2, \ldots \xi_q$ consisting of j elements. Similarly we can show that for $k \neq l$,

$$p(\omega_k, \omega_l|j \text{ of } \xi_1, \ldots \xi_q \in C_1) = \frac{1}{\pi^2}.$$

Therefore the variables $\omega_1, \omega_2, \ldots, \omega_j$ are mutually independent.

- *The eigenvalues $\phi_1, \phi_2, \ldots, \phi_{j-1}$ inside the unit circle but outside the circle of radius H are uniformly distributed and mutually independent:* Suppose $\omega_{i_1}, \omega_{i_2}, \ldots \omega_{i_{j-1}}$ with $i_1 < i_2 < \cdots < i_{j-1}$ are the eigenvalues inside the desired area. When we map ω_{i_k} to ϕ_k, the argument above applies to prove the uniformity and mutual independence of the variables $\phi_1, \phi_2, \ldots, \phi_{j-1}$.

- *Given c eigenvalues $\vartheta_1, \vartheta_2, \ldots \vartheta_c$ inside the left circle with radius $\frac{1-H}{2}$ centered at $(\frac{-(1+H)}{2}, 0)$, each eigenvalue is uniformly distributed and mutually independent:* This again follows from the arguments above by mapping ϕ_{i_k} to ϑ_k where $\phi_{i_1}, \phi_{i_2}, \ldots \phi_{i_c}$ are the eigenvalues inside the desired region with $i_1 < i_2 < \cdots < i_c$.

- *Assuming that the number of eigenvalues contained in the circle of radius D is fixed, the expected number of iterations by the algorithm does not depend on the radius D:* Consider the variables $\hat{\omega}_1$ denoting the locations of the j eigenvalues all inside the circle of radius D_1 and $\hat{\omega}_2 = \frac{D_2\hat{\omega}_1}{D_1}$ denoting the locations of the j eigenvalues inside the circle of radius D_2. Let us denote the number of iterations by Algorithm 7 with input $\hat{\omega}_1$ over the interval $[-D_1, D_1]$ by $X_1(\hat{\omega}_1)$ and the number of iterations with input $\hat{\omega}_2$ over the interval $[-D_2, D_2]$ by $X_2(\hat{\omega}_2)$. It immediately follows that $X_1(\hat{\omega}_1) = X_1(\frac{D_1\hat{\omega}_2}{D_2}) = X_2(\hat{\omega}_2)$. By exploiting this equality we can deduce

$E(X_1|N_1 = j) = E(X_2|N_2 = j)$,

$$\begin{aligned} E(X_1|N_1 = j) &= \int_{C_{D_1}} X_1(\hat{\omega}_1)p(\hat{\omega}_1)\, d\hat{\omega}_1 \\ &= \left(\frac{1}{\pi D_1^2}\right)^j \int_{C_{D_1}} X_1(\hat{\omega}_1)\, d\hat{\omega}_1 \\ &= \left(\frac{1}{\pi D_1^2}\right)^j \int_{C_{D_2}} X_1\left(\frac{D_1\hat{\omega}_2}{D_2}\right) \frac{D_1^{2j}}{D_2^{2j}} d\hat{\omega}_2 \\ &= \int_{C_{D_2}} X_2(\hat{\omega}_2)p(\hat{\omega}_2)\, d\hat{\omega}_2 \\ &= E(X_2|N_2 = j), \end{aligned}$$

where N_1 and N_2 are the number of eigenvalues inside the circle of radius C_{D_1} of radius D_1 and the circle of radius C_{D_2} of radius D_2, respectively. Note that the eigenvalues inside both the circle C_{D_1} and the circle C_{D_2} are uniformly distributed and independent as we discussed above.

By combining these remarks we obtain the equality $E(X|N = j) = E(X_l|N_l = j)$, since the eigenvalues are uniformly distributed and independent inside the circles and the sizes of the circles do not affect the expected number of iterations given that there are j eigenvalues inside the circles.

The next theorem establishes a recurrence equation for $E(X|N = j)$ in terms of $E(X|N = k)$, $k = 0 \ldots j-1$. Using the recurrence equation we will show that $E(X|N = j) = O(\sqrt{j})$ by induction. For convenience let us use the shorthand notation $E_j(X)$ for $E(X|N = j)$.

Theorem 28. *Suppose that the eigenvalues of the input matrices of size q are chosen from a uniform distribution independently inside the circle of radius μ and Algorithm 7 is run over the interval $[-1, 1]$. The quantity $E_j(X)$ can be characterized by the recurrence equation*

$$E_0(X) = 1 \tag{4.34}$$

and for all $0 < j < q$

$$E_j(X) = 2F_{j-1}(X) + 1, \tag{4.35}$$

where $F_{j-1}(X)$ is a linear combination of the expectations $E_0(X), \ldots E_{j-1}(X)$,

$$F_{j-1}(X) = \int_0^1 \left(\sum_{k=0}^{j-1} E_k(X) g_l(N_l = k|N = j, H = \beta)\right) h(H = \beta|N = j)\, d\beta. \tag{4.36}$$

X	:	Number of iterations performed by Algorithm 7.
N	:	Number of eigenvalues lying inside the unit circle.
H	:	Modulus of the eigenvalue closest to the origin.
X_l	:	Number of iterations performed by Algorithm 7 on the left interval $[-1, -H]$.
X_r	:	Number of iterations performed by Algorithm 7 on the right interval $[H, 1]$.
N_l	:	Number of eigenvalues lying inside the left circle centered at $-(1 + H)/2$ with radius $(1 - H)/2$.
$h(H\|N = j)$	:	The probability density function of the variable H given there are j eigenvalues inside the unit circle.
$g_l(N_l\|N = j, H = \beta)$	:	The probability density function of the variable N_l given there are j eigenvalues inside the unit circle and the smallest of the moduli of the eigenvalues is β.

Table 4.1: Notation for Theorem 28

Proof. Equation (4.34) is trivial; when there is no eigenvalue inside the unit circle, the algorithm will converge to an eigenvalue on or outside the unit circle and terminate.

For $j > 0$ at the first iteration of the algorithm, we compute the closest eigenvalue to the midpoint and repeat the same procedure with the left interval and with the right interval, so the equality

$$X = X_l + X_r + 1$$

and therefore the equality

$$E_j(X) = E(X_l|N = j) + E(X_r|N = j) + 1 \tag{4.37}$$

follows. Clearly the number of iterations on the left interval and on the right interval depend on the modulus of the computed eigenvalue. By the definition of conditional expectations, we deduce

$$E(X_l|N = j) = \int_0^1 E(X_l|N = j, H = \beta)h(H = \beta|N = j)\, d\beta \tag{4.38}$$

and similarly

$$E(X_r|N = j) = \int_0^1 E(X_r|N = j, H = \beta)h(H = \beta|N = j)\, d\beta. \tag{4.39}$$

Now we focus on the procedures applied on the left and right intervals. Let the modulus of the eigenvalue computed at the first iteration be β. There may be up to $j - 1$ eigenvalues inside the circle centered at the midpoint of the left interval $[-1, \beta]$ and with radius $\frac{1-\beta}{2}$. The expected number of iterations on the left interval is independent of the radius $\frac{1-\beta}{2}$ and the number of eigenvalues lying outside this circle. Therefore given the number of eigenvalues inside this circle, by the definition of conditional expectations, the equality

$$\begin{aligned} E(X_l|N = j, H = \beta) &= \sum_{k=0}^{j-1} E(X_l|N_l = k, N = j, H = \beta)g_l(N_l = k|N = j, H = \beta) \\ &= \sum_{k=0}^{j-1} E(X_l|N_l = k)g_l(N_l = k|N = j, H = \beta) \\ &= \sum_{k=0}^{j-1} E_k(X)g_l(N_l = k|N = j, H = \beta) \end{aligned} \tag{4.40}$$

is satisfied. A similar argument applies to the right interval to show the analogous equality

$$E(X_r|N=j,H=\beta)=\sum_{k=0}^{j-1}E_k(X)g_l(N_l=k|N=j,H=\beta). \tag{4.41}$$

By substituting (4.40) into (4.38), (4.41) into (4.39) and combining these with (4.37), we deduce the result.

□

Corollary 29 (Average case for Algorithm 7)**.** *Suppose that the eigenvalues of the matrices input to Algorithm 7 are selected uniformly and independently inside the circle of radius μ. The expectation $E_j(X)$ is bounded above by $c\sqrt{j+f}-1$ for all $c\geq\sqrt{12}$ and $f\in[4/c^2,1/3]$.*

Proof. The proof is by induction. In the base case, when there is no eigenvalue inside the unit circle, the algorithm iterates only once, *i.e.* $E_0(X)=1\leq c\sqrt{f}-1$.

Assume that for all $k<j$, the claim $E_k(X)\leq c\sqrt{k+f}-1$ holds. We need to show that the inequality $E_j(X)\leq c\sqrt{j+f}-1$ is satisfied under this assumption. By definition (4.36) in Theorem 28 we have

$$F_{j-1}(X)\leq\int_0^1\left(\sum_{k=0}^{j-1}(c\sqrt{k+f}-1)g_l(N_l=k|N=j,H=\beta)\right)h(H=\beta|N=j)d\beta. \tag{4.42}$$

As we argued before, the uniformity and independence of each of the $j-1$ eigenvalues inside the unit circle but outside the circle of radius $H=\beta$ are preserved. In other words $g_l(N_l|N=j,H=\beta)$ is a binomial density function and we can explicitly write $g_l(N_l=k|N=j,H=\beta)$, the probability that there are k eigenvalues inside the left circle given that there are $j-1$ eigenvalues contained in the unit circle and outside the circle of radius β, as

$$g_l(N_l=k|N=j,H=\beta)=\binom{j-1}{k}\left(\frac{1-\beta}{4(1+\beta)}\right)^k\left(1-\frac{1-\beta}{4(1+\beta)}\right)^{j-1-k}.$$

Now the expected value of the binomial distribution above is $(j-1)\frac{1-\beta}{4(1+\beta)}$. From

Lemma 27, we deduce

$$\begin{aligned}
\frac{\sqrt{j+f}}{2} &\geq \frac{\sqrt{j-1+4f}}{2} \\
&\geq \sqrt{\frac{(1-\beta)(j-1)}{4(1+\beta)}+f} \\
&= \sqrt{\sum_{k=0}^{j-1}(k+f)g_l(N_l=k|N=j,H=\beta)} \\
&> \sum_{k=0}^{j-1}\sqrt{k+f}\; g_l(N_l=k|N=j,H=\beta).
\end{aligned}$$

Substituting the upper bound $\frac{\sqrt{j+f}}{2}$ for $\sum_{k=0}^{j-1}\sqrt{k+f}\; g_l(N_l=k|N=j,H=\beta)$ in (4.42) yields

$$F_{j-1}(X) \leq \int_0^1 (\frac{c\sqrt{j+f}}{2}-1)h(H=\beta|N=j)\,d\beta = \frac{c\sqrt{j+f}}{2}-1. \qquad (4.43)$$

Now it follows from (4.35) that

$$E_j(X) \leq c\sqrt{j+f}-1 \qquad (4.44)$$

as desired.

□

Recall that we intend to apply the divide-and-conquer approach to $\mathcal{A}$ which has size $2n^2 \times 2n^2$. Assume that the conditions of Corollary 29 hold for the eigenvalues of $\mathcal{A}$ and the circle of radius D contains all of the eigenvalues. Suppose also that for any shift ν, convergence of the shifted inverse iteration or shift-and-invert Arnoldi method to the closest eigenvalue requires the matrix vector multiplication $(\mathcal{A}-\nu I)^{-1}u$ for various u only a constant number of times. Then the average running time of each trisection step is $O(n^4)$, since finding the closest eigenvalue takes $O(n^3)$ time (which is the cost of solving a Sylvester equation of size $2n$ a constant number of times) and we compute the closest eigenvalue $O(n)$ times at each trisection step on average. Because of the special structure of the Kronecker product matrix $\mathcal{A}$, even if the input matrices have eigenvalues uniformly distributed and mutually independent, the eigenvalues of $\mathcal{A}$ may not have this property. However, the numerical examples in the next subsection suggest that the number of closest eigenvalue computations as a function of the size of the Kronecker product matrices is still bounded by $O(\sqrt{q})$. According to Theorem 26 in the worst-case scenario, each trisection step requires $O(n^5)$ operations, which is an improvement over computing all of the eigenvalues of $\mathcal{A}$.

4.3.4 Further remarks

The divide-and-conquer approach requires an upper bound on the norm of $\mathcal{A}$. In practice this parameter may be set arbitrarily large and the efficiency of the algorithm is affected insignificantly. Alternatively, the upper bound on $\|\mathcal{A}\|$ in (4.53) derived below can be used.

Improvements to the divide-and-conquer approach seem possible. As the upper and lower bounds become closer, the Kronecker product matrices $\mathcal{A}$ in two successive iterations differ only slightly. Therefore it is desirable to benefit from the eigenvalues computed in the previous iteration in the selection of the shifts. We address further details of the new algorithm below.

Sylvester equation solvers

The Sylvester equations needed to perform the multiplication $(\mathcal{A} - \nu I)^{-1}u$ are not sparse in general. We solve them by first reducing the coefficient matrices on the left-hand side of (4.32) to upper quasi-triangular forms (block upper triangular matrices with 1×1 and 2×2 blocks on the diagonal). Then the algorithm of Bartels and Stewart can be applied [3]. In our implementation we used the LAPACK routine `dtrsyl` [1] which is similar to the method of Bartels and Stewart, but rather than computing the solution column by column it generates the solution row by row, bottom to top. A more efficient alternative may be the recursive algorithm of Jonsson and Kågström [41].

Incorporating BFGS

By combining the new verification scheme and BFGS, it is possible to come up with a more efficient and accurate algorithm. A local minimum of the function $\sigma_{\min}([A - \lambda I \quad B])$ can be found in a cheap manner by means of the BFGS optimization algorithm. Notice that the cost of this local optimization step is $O(1)$, since we are searching over two unknowns, namely, the real and the imaginary parts of λ. Using the new verification scheme we can check whether the local minimum is indeed a global minimum as described in [13, Algorithm 5.3]. If the local minimum is not a global minimum, the new verification scheme also provides us with a point λ' where the value of the function $\sigma_{\min}([A-\lambda I \;\; B])$ is less than the local minimum. Therefore we can repeat the application of BFGS followed by the new scheme until we verify that the local minimum is a global minimum.

Alternative eigenvalue problem

To see whether there exists an α such that $D(\alpha, \delta)$ and $D(\alpha + \eta, \delta)$ share an eigenvalue, we extract the real eigenvalues of $\mathcal{A}$. Alternatively we can solve the generalized eigenvalue problem $(\mathcal{F}(\delta, \eta) - \lambda\mathcal{G})x = 0$ defined in (4.21). The real eigenvalue extraction techniques are applicable to this problem as well, since the scalar λ is an eigenvalue of the pencil $\mathcal{F}(\delta, \eta) - \lambda\mathcal{G}$ if and only if $\frac{1}{\lambda - \nu}$ is an eigenvalue of the matrix $(\mathcal{F}(\delta, \eta) - \nu\mathcal{G})^{-1}\mathcal{G}$. The multiplication $x = (\mathcal{F}(\delta, \eta) - \nu\mathcal{G})^{-1}\mathcal{G}y$ can be performed efficiently by solving the linear system $(\mathcal{F}(\delta, \eta) - \nu\mathcal{G})x = \mathcal{G}y$. When we write this linear system in matrix form, we obtain the Sylvester equation (4.19) but with α replaced by ν and with the matrix

$$\begin{pmatrix} -2Y_{11} & 0 \\ 0 & 2Y_{22} \end{pmatrix}$$

replacing 0 on the righthand side, where $y = [\mathbf{vec}(Y_{11})\ y_{12}\ y_{21}\ \mathbf{vec}(Y_{22})]^T$ with equal-sized block components. Notice that the fact that the eigenvalue problem $(\mathcal{F}(\delta, \eta) - \lambda\mathcal{G})x = 0$ is of double size compared to the eigenvalue problem $\mathcal{A}x = \lambda x$ is not an efficiency concern, since we still solve Sylvester equations of the same size. The real issue is that these two eigenvalue problems have different conditioning. Theoretically either of them can be better conditioned than the other in certain situations. In practice we retrieved more accurate results with the eigenvalue problem $\mathcal{A}x = \lambda x$ most of the time, even though there are also examples on which the algorithm using $(\mathcal{F}(\delta, \eta) - \lambda\mathcal{G})x = 0$ yields more accurate results.

Accuracy issues

In general neither the old method based on Gu's verification scheme nor the new method is stable as both of them require the computation of the eigenvalues of matrices with large norm. Gu's method in [31] suffers from the fact that the matrix Q_{12} in (4.17) becomes highly ill-conditioned as $\delta \to 0$ and is not invertible at the limit. Therefore the method is not appropriate for accurate computation for uncontrollable or nearly uncontrollable pairs.

For the new method based on the real eigenvalue extraction technique, the accuracy trouble is caused by the fact that the norm of $\mathcal{A}$ blows up as η goes to 0. There are two potential problems with computing eigenvalues of matrices with large norm. The first one is that for any backward stable eigenvalue solver used, a computed eigenvalue of $\mathcal{A}$ differs from the exact one typically by a quantity with modulus on the order of

$$\frac{\|\mathcal{A}\|\epsilon_{mach}}{|w^*z|}$$

where w and z are the corresponding unit left and right eigenvectors. The second problem is related to the fact that we compute the eigenvalues of $\mathcal{A}$ iteratively and therefore it is necessary to solve the linear system $(\mathcal{A}-\nu I)x = u$ for various ν and u. Unfortunately, the absolute error for the solution of the linear system depends on the condition number $\kappa(\mathcal{A}-\nu I) = \|(\mathcal{A}-\nu I)\|\|(\mathcal{A}-\nu I)^{-1}\|$. It is known that if ν is close to an eigenvalue of A, then the fact that the matrix $(\mathcal{A}-\nu I)$ is close to singularity does not cause numerical trouble [63, Chapter 4]. However, in our case the norm of $(\mathcal{A}-\nu I)$ is also large. In practice we observe that this affects the convergence of Arnoldi's method; there are cases, especially when the norm of $\mathcal{A}$ is large, where the QR algorithm computes eigenvalues accurately while `eigs` fails to converge. In our experience `eigs` has convergence problems typically when the norm of $\mathcal{A}$ reaches the order of 10^{10}.

In what follows we gain insight into the variation in the norm of $\mathcal{A}$ as η decreases to zero. What makes $\mathcal{A}$ ill-conditioned is the inverted matrix in (4.23) whose norm is equal to $1/\sigma_{\min}(I \otimes A - (A-\eta I)^T \otimes I)$. Let us define

$$\mathcal{K}(\eta) = (I \otimes A - (A-\eta I)^T \otimes I).$$

In the limit as $\eta \to 0$, $\mathcal{K}(0)$ is singular. Suppose that the eigenvalues of A are nondefective (each eigenvalue has its algebraic multiplicity equal to its geometric multiplicity) and are ordered as $\lambda_1 < \lambda_2 < \cdots < \lambda_s$ in strictly ascending order. For each eigenvalue $\lambda_j,\ j = 1, \ldots, s$, with the multiplicity m_j, suppose that an orthonormal basis for the associated right eigenspace is

$$X_j = [x_{j,1}\ x_{j,2}\ \ldots\ x_{j,m_j}]$$

and an orthonormal basis for the associated left eigenspace is

$$Y_j = [y_{j,1}\ y_{j,2}\ \ldots\ y_{j,m_j}].$$

Note that X' is a solution of the Sylvester equation $AX - XA = 0$ if and only if the equalities

$$\mathcal{K}(0)\mathbf{vec}(X') = 0 \ \text{ and } \ \mathbf{rvec}(X')\mathcal{K}(0) = 0$$

hold where $\mathbf{rvec}(X)$ denotes the row vector obtained by concatenating the rows of the matrix X. It immediately follows that for all $j = 1, \ldots, s$, right eigenvectors $x_{j,l},\ l = 1, \ldots, m_j$ and left eigenvectors $y_{j,r},\ r = 1, \ldots, m_j$, the vector $\mathbf{vec}(x_{j,l}y_{j,r}^*) = \bar{y}_{j,r} \otimes x_{j,l}$ lies in the right null space of $\mathcal{K}(0)$, while the vector $(\mathbf{rvec}(x_{j,l}y_{j,r}^*))^* = \bar{x}_{j,l} \otimes y_{j,r}$ lies in the left null space of $\mathcal{K}(0)$. Indeed a unitary matrix whose columns form an orthonormal basis for the right null-space of $\mathcal{K}_0$ is

$$V = [V_1\ V_2\ \ldots\ V_s] \tag{4.45}$$

where for $j = 1, \ldots, s$ the columns of the matrix $V_j \in \mathbb{C}^{n^2 \times m_j^2}$ are the Kronecker products of the left and right eigenvectors of A associated with λ_j,

$$V_j = [\bar{y}_{j,1} \otimes x_{j,1} \;\; \ldots \;\; \bar{y}_{j,m_j} \otimes x_{j,1} \;\; \bar{y}_{j,1} \otimes x_{j,2} \;\; \ldots \;\; \bar{y}_{j,m_j} \otimes x_{j,m_j}]. \tag{4.46}$$

Using the property that the right eigenvectors $x_{j,l}$ and $x_{r,d}$ and the left eigenvectors $y_{j,l}$ and $y_{r,d}$ are orthogonal to each other unless $j = r$ and $l = d$, it is straightforward to deduce that $V_j^* V_j = I$ and $V_j^* V_l = 0$ for $j \neq l$ and therefore V is unitary. Similarly, a unitary matrix whose columns form an orthonormal basis for the left null space of $\mathcal{K}(0)$ is

$$U = [U_1 \; U_2 \; \ldots \; U_s] \tag{4.47}$$

with the unitary blocks

$$U_j = [\bar{x}_{j,1} \otimes y_{j,1} \;\; \ldots \;\; \bar{x}_{j,m_j} \otimes y_{j,1} \;\; \bar{x}_{j,1} \otimes y_{j,2} \;\; \ldots \;\; \bar{x}_{j,m_j} \otimes y_{j,m_j}] \tag{4.48}$$

for $j = 1, \ldots, s$. Notice also that the property $U_j^* U_l = 0$ holds for $j \neq l$. Having identified the left and right null space of $\mathcal{K}(0)$, we derive the first order variation in the minimum singular value of $\mathcal{K}(\eta)$ as a function of η around zero.

Theorem 30. *Suppose that the eigenvalues of A are nondefective and V_j and U_j for $j = 1, \ldots, n$ are defined by (4.46) and (4.48), respectively. The minimum singular value of $K(\eta)$ satisfies*

$$\sigma_{\min}(K(\eta)) = \eta \delta_\sigma + o(\eta), \tag{4.49}$$

where

$$\delta_\sigma = \min_{1 \leq j \leq s} \sigma_{\min}(U_j^* V_j). \tag{4.50}$$

Proof. The set of eigenvalues of the Hermitian matrix

$$\mathcal{D}(\eta) = \begin{bmatrix} 0 & (\mathcal{K}(\eta))^* \\ \mathcal{K}(\eta) & 0 \end{bmatrix}$$

is the set of singular values of $\mathcal{K}(\eta)$ with positive and negative signs. In particular for $\mathcal{D}(0)$ we have the eigenvalue decomposition

$$\mathcal{D}(0) = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} V_c & V_c & V & V \\ U_c & -U_c & U & -U \end{bmatrix} \right) \begin{bmatrix} \Sigma & 0 & 0 & 0 \\ 0 & -\Sigma & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} V_c^* & U_c^* \\ V_c^* & -U_c^* \\ V^* & U^* \\ V^* & -U^* \end{bmatrix} \right),$$

where V_c and U_c are matrices with orthonormal columns spanning the null spaces of V (defined by (4.45)) and U (defined by (4.47)), respectively. We are

interested in the change in the smallest eigenvalue (which is equal to zero) of the Hermitian matrix $\mathcal{D}(0)$ with multiplicity $d_s = 2\sum_{j=1}^{s} m_j^2$ under the perturbation

$$\delta\mathcal{D} = \frac{\mathcal{D}(\eta) - \mathcal{D}(0)}{\eta} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}.$$

For sufficiently small η each of the smallest d_s eigenvalues of $\mathcal{D}(\eta)$ in magnitude can be expressed in the form

$$\mu_l(\eta) = \lambda_{\min}(\mathcal{D}(0)) + a_l\eta + o(\eta) = a_l\eta + o(\eta) \tag{4.51}$$

for $l = 1, \ldots, d_s$ where each a_l correspond to a distinct eigenvalue of the matrix

$$\left(\frac{1}{2}\begin{bmatrix} V & V \\ U & -U \end{bmatrix}\begin{bmatrix} V^* & U^* \\ V^* & -U^* \end{bmatrix}\right)\delta\mathcal{D}\left(\frac{1}{2}\begin{bmatrix} V & V \\ U & -U \end{bmatrix}\begin{bmatrix} V^* & U^* \\ V^* & -U^* \end{bmatrix}\right) = \begin{bmatrix} 0 & VV^*UU^* \\ UU^*VV^* & 0 \end{bmatrix}$$

(see [47, Theorem 7.9.1] and [60]). Therefore $|a_l|$ is a singular value of the matrix UU^*VV^*. Furthermore for $l \neq j$, since the vectors $x_{j,r}$ and $y_{l,d}$ are orthogonal to each other for all r and d, it immediately follows that $U_l^*V_j = 0$ and we obtain

$$UU^*VV^* = \sum_{j=0}^{s} U_jU_j^*V_jV_j^* = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_s \end{bmatrix}\begin{bmatrix} U_1^*V_1 & & \\ & \ddots & \\ & & U_s^*V_s \end{bmatrix}\begin{bmatrix} V_1^* & & \\ & \ddots & \\ & & V_s^* \end{bmatrix}.$$

Because of the properties that $U_j^*U_l = 0$ and $V_j^*V_l = 0$ for $l \neq j$, the leftmost and rightmost matrices in the rightmost product are unitary. Therefore each a_l in absolute value is a singular value of one of the matrices $U_j^*V_j,\ j = 1, \ldots, s$. We deduce that

$$\min_{1\leq l\leq d_s} |a_l| = \min_{1\leq j\leq s} \sigma_{\min}(U_j^*V_j).$$

By combining the above inequality with (4.51) and noting that

$$\sigma_{\min}(\mathcal{K}(\eta)) = \min_{l=1,\ldots,s} |\mu_l(\eta)| = \eta \min_{l=1,\ldots,d_s} |a_l| + o(\eta),$$

we complete the proof. □

Now the first-order change δ_σ in the equality (4.50) depends on the alignment of the left and right eigenspaces with respect to each other. If there exists an eigenvalue λ_j of A for which the left eigenspace and the right eigenspace are orthogonal or close to orthogonal, δ_σ is small, meaning that the smallest singular value of $\mathcal{K}(0)$ is very insensitive to perturbations. At the other extreme, if all

of the left and right eigenspaces are perfectly aligned, that is $V_j = U_j$ for all j, then $\delta_\sigma = 1$, that is the variation in the minimum singular value is maximized. When the multiplicities of all of the eigenvalues of A are one, the equality (4.50) simplifies considerably.

Corollary 31. *Suppose that for all $j = 1, \ldots, s$ the multiplicity of λ_j is one. Then the equality (4.49) holds for*

$$\delta_\sigma = \min_{1 \le j \le s} |y_j^* x_j|^2. \tag{4.52}$$

Proof. For the special case when all of the multiplicities are one, $V_j = \bar{y}_j \otimes x_j$ and $U_j = \bar{x}_j \otimes y_j$ for $j = 1, \ldots, s$, so (4.50) becomes

$$\delta_\sigma = \min_{1 \le j \le s} \sigma_{\min}((x_j^T \otimes y_j^*)(\bar{y}_j \otimes x_j)) = \min_{1 \le j \le s} |y_j^* x_j|^2.$$

□

According to Corollary 31 the absolute condition number of the worst conditioned eigenvalue of A determines the change in the minimum singular value, assuming that A has simple eigenvalues.

The norm of the inverted matrix in (4.23) is $\frac{1}{\eta\delta_\sigma}$ up to first-order terms where δ_σ is given by (4.50) in general or specifically, when the eigenvalues of A are simple, by (4.52). Using the triangle inequality and the Cauchy-Schwarz inequality, an approximate upper bound on $\mathcal{A}$ is given by

$$\|A\| + \frac{(\|BB^*/\delta - \delta I\| + \delta I)^2}{\eta\delta_\sigma}$$

which reduces to

$$\|A\| + \frac{(\|BB^*/\delta - \delta I\| + \delta I)^2}{\eta \min_{1 \le j \le s} |y_j^* x_j|^2} \tag{4.53}$$

when the eigenvalues of A are simple.

Notice that the upper bound given by (4.53) can be efficiently computed in $O(n^3)$ time and therefore in an implementation it can be used to estimate the length of the smallest interval containing the distance to uncontrollability that can possibly be computed accurately. Surprisingly, the norm of $\mathcal{A}$ heavily depends on the worst conditioned eigenvalue of A, but it has little to do with the norm of A. For instance, when A is normal and $\|B\|$ is not very large, we expect that $\|\mathcal{A}\|$ exceeds 10^{10} only when η is smaller than 10^{-10} unless the pair (A, B) is nearly uncontrollable. This in turn means that we can reliably compute an interval of length 10^{-10} containing the distance to uncontrollability. On the other hand, when A is far from being normal or the pair (A, B) is close to being uncontrollable and a small interval is required, the new method performs poorly.

4.4 Numerical examples

We first compare the accuracy of the three trisection algorithms, namely the new trisection algorithm for low precision (§4.2), the original high-precision algorithm of [13] based on Gu's verification condition in [31] and the new high-precision algorithm based on divide-and-conquer (§4.3) on a variety of examples. Secondly, we aim to show the asymptotic running time difference between the new method with the divide-and-conquer approach and Gu's method. Finally we illustrate that our theoretical result in §4.3.4 to estimate the norm of the Kronecker product matrices holds in practice.

4.4.1 Accuracy of the new algorithm and the old algorithm

We present results comparing the accuracy of the new methods for low precision and for high precision using the divide-and-conquer approach with the original trisection method in [13] based on Gu's verification condition in [31]. In exact arithmetic both the method in [13] and the new method using the divide-and-conquer approach must return the same interval, since they perform the same verification by means of different but equivalent eigenvalue problems. On the other hand, in the algorithm for low precision we verify which one of the inequalities (4.3) and (4.4) holds by means of a different method. In particular, when the inequalities $\delta_1 \geq \tau(A, B) > \delta_2$ hold simultaneously, the method for low precision may update the lower bound, while the methods for high precision update the upper bound, or vice versa. Therefore the intervals computed by the low-precision algorithm and the high-precision algorithms are not necessarily the same, but must overlap. Our data set consists of pairs (A, B) where A is provided by the software package *EigTool* [73] and B has entries selected independently from the normal distribution with zero mean and variance one. The data set is available on the web site [58]. In all of the tests the initial interval is set to $[0, \sigma_{\min}([A \quad B])]$. For the low-precision algorithm the trisection step is repeated until an interval $[l, u]$ with $u - l \leq 10^{-2}$ is obtained. For the high-precision algorithms the trisection step is repeated until an interval of length at most 10^{-4} is obtained.

When the second and third columns in Table 4.2 and 4.3 are considered, on most of the examples the methods return the same interval with the exception of the companion, Demmel, Godunov and gallery5 examples. The common property of these matrices is that they have extremely ill-conditioned eigenvalues. As we discussed in §4.3.4, when the matrix A has an ill-conditioned eigenvalue, the new method for high precision with the divide and conquer approach is not expected to produce accurate small intervals containing the distance to uncon-

Example	**New Method (High Prec.)**	**Original Trisection Method**	**New Method (Low Prec.)**
Airy(5,2)	(0.03759,0.03767]	(0.03759,0.03767]	(0.030,0.038]
Basor-Morrison(5,2)	(0.68923,0.68929]	(0.68923,0.68929]	(0.681,0.689]
Chebyshev(5,2)	(0.75026,0.75034]	(0.75026,0.75034]	(0.743,0.750]
Companion(5,2)	(0.42431,0.42438]	(0.42431,0.42438]	(0.416,0.425]
Convection Diffusion(5,2)	(0.69829,0.69836]	(0.69829,0.69836]	(0.690,0.699]
Davies(5,2)	(0.23170,0.23176]	(0.23170,0.23176]	(0.224,0.233]
Demmel(5,2)	(0.09090,0.09097]	(0.09049,0.09056]	(0.083,0.092]
Frank(5,2)	(0.45907,0.45916]	(0.45907,0.45916]	(0.452,0.459]
Gallery5(5,2)	(0.17468,0.17474]	(0.02585,0.02592]	(0.021,0.030]
Gauss-Seidel(5,2)	(0.06279,0.06288]	(0.06279,0.06288]	(0.056,0.064]
Grcar(5,2)	(0.49571,0.49579]	(0.49571,0.49579]	(0.491,0.498]
Hatano(5,2)	(0.39570,0.39578]	(0.39570,0.39578]	(0.391,0.398]
Kahan(5,2)	(0.18594,0.18601]	(0.18594,0.18601]	(0.178,0.187]
Landau(5,2)	(0.41766,0.41773]	(0.41766,0.41773]	(0.410,0.419]
Orr-Sommerfield(5,2)	(0.04789,0.04796]	(0.04789,0.04796]	(0.041,0.050]
Supg(4,2)	(0.06546,0.06554]	(0.06546,0.06554]	(0.059,0.066]
Transient(5,2)	(0.11027,0.11036]	(0.11027,0.11036]	(0.104,0.112]
Twisted(5,2)	(0.14929,0.14936]	(0.14929,0.14936]	(0.143,0.153]

Table 4.2: For pairs (A, B) with A chosen from *EigTool* as listed above in the leftmost column and B normally distributed, intervals $(l, u]$ that are supposed to contain the distance to uncontrollability for the system (A, B) are computed by three different trisection algorithms with $u - l \leq 10^{-4}$ for the high precision algorithms and $u - l \leq 10^{-2}$ for the low precision algorithm. The size of the system (n, m) is provided in parentheses in the leftmost column.

Example	**New Method (High Prec.)**	**Original Trisection Method**	**New Method (Low Prec.)**
Airy(10,4)	(0.16337,0.16345]	(0.16337,0.16345]	(0.158,0.168]
Basor-Morrison(10,4)	(0.60974,0.60980]	(0.60974,0.60980]	(0.604,0.613]
Chebyshev(10,4)	(0.82703,0.82711]	(0.82703,0.82711]	(0.819,0.829]
Companion(10,4)	(0.46630,0.46637]	(0.46610,0.46616]	(0.459,0.468]
Convection Diffusion(10,4)	(1.48577,1.48586]	(1.48577,1.48586]	(0.479,0.487]
Davies(10,4)	(0.70003,0.70012]	(0.70003,0.70012]	(0.695,0.703]
Demmel(10,4)	(0.12049,0.12057]	(0.11998,0.12006]	(0.113,0.121]
Frank(10,4)	(0.67405,0.67414]	(0.67405,0.67414]	(0.666,0.674]
Gauss-Seidel(10,4)	(0.05060,0.05067]	(0.05060,0.05067]	(0.046,0.055]
Godunov(7,3)	(1.23802,1.23810]	(1.23764,1.23773]	(1.233,1.240]
Grcar(10,4)	(0.44178,0.44185]	(0.44178,0.44185]	(0.434,0.444]
Hatano(10,4)	(0.23297,0.23304]	(0.23297,0.23304]	(0.227,0.236]
Kahan(10,4)	(0.05587,0.05594]	(0.05587,0.05594]	(0.049,0.058]
Landau(10,4)	(0.28166,0.28174]	(0.28166,0.28174]	(0.275,0.285]
Markov Chain(6,2)	(0.04348,0.04358]	(0.04348,0.04358]	(0.035,0.044]
Markov Chain(10,4)	(0.07684,0.07693]	(0.07684,0.07693]	(0.070,0.078]
Orr-Sommerfield(10,4)	(0.07836,0.07843]	(0.07836,0.07843]	(0.071,0.080]
Skew-Laplacian(8,3)	(0.01001,0.01011]	(0.01001,0.01011]	(0.004,0.013]
Supg(9,4)	(0.03627,0.03634]	(0.03627,0.03634]	(0.029,0.037]
Transient(10,4)	(0.13724,0.13731]	(0.13724,0.13731]	(0.131,0.140]
Twisted(10,4)	(0.77178,0.77185]	(0.77178,0.77185]	(0.764,0.774]

Table 4.3: Comparison of trisection methods on larger pairs.

trollability. One false conclusion that one may draw from Table 4.2 and 4.3 is that the original trisection method is always more accurate than the new method with the divide-and-conquer approach. Indeed for the Basor-Morrison, Grcar or Landau examples with $n = 5$ (for which the eigenvalues are fairly well-conditioned) the new method with the divide and conquer approach generates more accurate results than Gu's method when one seeks intervals of length around 10^{-6}. In terms of accuracy these two methods have different weaknesses. The algorithm for low precision is the most accurate one among all three. Unfortunately, it is well-suited only to retrieve a rough estimate of the distance to uncontrollability.

4.4.2 Running times of the new algorithm with the divide-and-conquer approach on large matrices

To observe the running time differences between the original trisection algorithm based on Gu's verification scheme and the new trisection method with the divide-and-conquer approach, we run the algorithms on pairs (A, B) of various size, where A is a Kahan matrix available through *EigTool* and B is a normally distributed matrix. We normalize the pairs (by dividing them by $\sigma_{\min}([A\ B])$) so that the same number of trisection steps are required. For $(n, m) = (40, 24)$ we did not run the trisection algorithm with Gu's verification scheme, since it takes an excessive amount of time. Instead, we extrapolated its running time. For all other sizes, both of the methods return the same interval of length approximately 10^{-4}. In Table 4.4 the running times of both of the algorithms and the average number of calls to `eigs` made by the divide-and-conquer approach are provided for various sizes. For small pairs Gu's old verification scheme is faster. However, for matrices of size 20 and larger the new method with the divide-and-conquer approach is more efficient and the difference in the running times increases drastically as a function of n. In the third column the average number of calls to `eigs` is shown and apparently varies linearly with n. Figure 4.4 displays plots of the running times as functions of n using a log scale. The asymptotic difference in the running times agrees with the plots.

4.4.3 Estimating the minimum singular values of the Kronecker product matrices

In §4.3.4 we derived the first-order change that reduces to the formula

$$\sigma_{\min}(\mathcal{K}(\eta)) = \sigma_{\min}(A \otimes I - (A - \eta I)^T \otimes I) = \min_{1 \le j \le s} |y_j^* x_j|^2 \eta + o(\eta)$$

Size (n,m)	t_{cpu} **(Gu's Method)**	t_{cpu} **(New Method)**	**no. of calls to eigs**
(10,6)	47	171	34
(20,12)	3207	881	63
(30,18)	46875	3003	78
(40,24)	263370	7891	92

Table 4.4: Running times of Gu's method and the new method with the divide and conquer approach in seconds and the average number of calls to `eigs` by the new method for Kahan-random matrix pairs of various size.

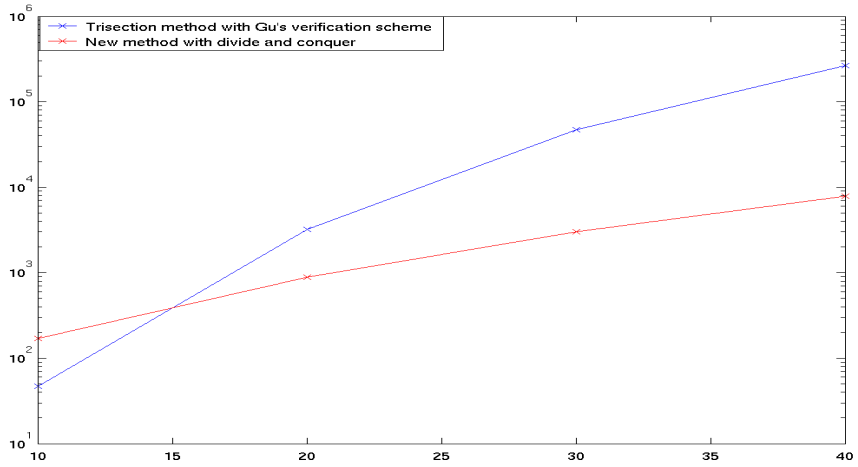

Figure 4.4: Running times of the methods on Kahan-random matrix pairs are displayed as functions of the size of the matrix in logarithmic scale.

size	$\hat{\delta}_\sigma$	$\tilde{\delta}_\sigma$
5	0.5195	0.6527
10	0.0285	0.0426
15	0.0009	0.0016
20	2.172×10^{-5}	4.870×10^{-5}

Table 4.5: Comparison of the quantities $\hat{\delta}_\sigma$ and $\tilde{\delta}_\sigma$ defined by (4.54) for the Grcar matrices of various size.

size	$\hat{\delta}_\sigma$	$\tilde{\delta}_\sigma$
5	0.0023	0.0038
10	1.355×10^{-6}	2.162×10^{-6}
15	2.430×10^{-9}	4.708×10^{-9}
20	2.803×10^{-11}	1.816×10^{-11}

Table 4.6: Comparison of the quantities $\hat{\delta}_\sigma$ and $\tilde{\delta}_\sigma$ for the Kahan matrices of various size.

under the assumption that all of the eigenvalues of A are simple, where y_j and x_j are the unit left and right eigenvectors corresponding to the jth eigenvalue. To illustrate the accuracy of the first order change we derived in practice, we compare the quantities

$$\hat{\delta}_\sigma = \frac{\sigma_{\min}(\mathcal{K}(10^{-6}))}{10^{-6}}, \qquad \tilde{\delta}_\sigma = \min_{1 \le j \le s} |y_j^* x_j|^2. \tag{4.54}$$

for the Grcar and Kahan matrices of various size available through *EigTool*. Table 4.5 and 4.6 show that these quantities for the Grcar matrices and Kahan matrices are within a factor of three of each other. We performed tests on various other examples and have not found an example for which one of the two quantities above is more than three times the other.

Chapter 5

Distance to Uncontrollability for Higher-Order Systems

In this chapter we focus on the problem of computing the distance to the closest uncontrollable system for the higher-order system

$$K_k x^{(k)}(t) + \cdots + K_1 x'(t) + K_0 x(t) = Bu(t) \tag{5.1}$$

defined by (1.19). For the special case of the descriptor system (for which the characterization of the controllability was given by Chu [19, 20])

$$Ex'(t) = Ax(t) + Bu(t),$$

Byers has studied the distance to uncontrollability and provided a generalization of the characterization (4.2) in [17]. We are not aware of the existence of any other work on the distance to uncontrollability for higher-order systems. We have also discussed the work presented in this chapter in [57].

In the next section we provide a singular-value minimization characterization for the definition (1.19). We will see that the definitions (1.19) in the spectral norm and the Frobenius norm are equivalent, just as in the first-order case, and the characterization we derive reduces to the Eising characterization (4.2) for the first-order system. In §5.2 we describe a trisection algorithm locating the global minimum of the associated optimization problem. The generalizations of the bisection algorithm of [31], the trisection algorithm of [13] or the trisection algorithm based on the real eigenvalue extraction technique in the previous chapter are too expensive to be practical. We present an algorithm that has similarities with the low-precision approximation technique that is suggested in the previous chapter for the first-order system. The first few steps of the new algorithm are comparatively cheap, but as we require more accuracy the algorithm becomes computationally intensive. With a complexity of $O\left(\frac{1}{\arccos(1-(\frac{tol}{k})^2)} n^3 k^4\right)$

with *tol* denoting the accuracy we require, it is devised for a few digits of precision. The algorithm depends on the extraction of the imaginary eigenvalues of $*$-even matrix polynomials of size $2n$ and degree $2k$. (See §A.3 in the appendix for how to solve $*$-even polynomial eigenvalue problems while preserving the symmetry of the spectrum.) §5.3 is devoted to numerical examples illustrating the efficiency of the algorithm. Note that throughout this chapter we usually use $\|\cdot\|$ for either the spectral or the Frobenius norm interchangeably when the results hold for both of the norms or when the type of the norm is clear from the context. At other times we clarify the choice of norm using the notation $\|\cdot\|_2, \|\cdot\|_F$ for the spectral and the Frobenius norm, respectively.

5.1 Properties of the higher-order distance to uncontrollability and a singular value characterization

The set of controllable tuples is clearly a dense subset of the whole space of matrix tuples. But this does not mean that the uncontrollable tuples are isolated points. On the contrary, there are uncontrollable subspaces. For instance, the system (5.1) with $K_0 = 0$ and rank $B < n$ is uncontrollable for all $K_k, \ldots, K_1$. Therefore we shall first see that $\tau(P, B, \gamma)$ is indeed attained at some $(\Delta K_k, \ldots, \Delta K_0, \Delta B)$.

Lemma 32. *There exists an uncontrollable tuple* $(K_k + \gamma_k \Delta K_k, \ldots, K_0 + \gamma_0 \Delta K_0, B + \Delta B)$ *such that* $\tau(P, B, \gamma) = \|\Delta K_k \quad \ldots \quad \Delta K_0 \quad \Delta B\|$ *and* $\|\Delta K_j\| \leq \gamma_j \|B\|$ *for all* j, $\|\Delta B\| \leq \|B\|$.

Proof. The matrix $[P(\lambda) \;\; 0]$ is rank deficient at the eigenvalues of P. Therefore $\tau(P, B, \gamma) \leq \|B\|$, meaning that we can restrict the perturbations to the ones satisfying $\|\Delta K_j\| \leq \gamma_j \|B\|$ and $\|\Delta B\| \leq \|B\|$.

Furthermore the set of uncontrollable tuples is closed. To see this, consider any sequence $\{(K'_k, \ldots, K'_0, B')\}$ of uncontrollable tuples. Now for any tuple in the sequence define the associated polynomial as $P'(\lambda) = \sum_{j=0}^{k} \lambda^j K'_j$. The matrix $[P'(\lambda) \;\; B]$ is rank-deficient for some λ, so all combinations of n columns of this matrix are linearly dependent. Let us denote the $l = \begin{pmatrix} m+n \\ n \end{pmatrix}$ polynomials associated with the determinants of the combinations of n columns by $p_1(\lambda), p_2(\lambda), \ldots, p_l(\lambda)$ in any order. These polynomials must share a common root; otherwise $[P'(\lambda) \;\; B]$ would not be rank-deficient for some λ. The common roots $r_1, r_2, \ldots, r_l$ are continuous functions of the tuple $\{(K'_k, \ldots, K'_0, B')\}$,

which means that at any cluster point of the sequence, $r_1 = r_2 = r_3 = \cdots = r_l$. This shows that the set is closed.

Since we are minimizing the spectral or the Frobenius norm over a compact set, $\tau(P, B, \gamma)$ must be attained at some $\|\Delta K_k \ \ldots \ \Delta K_0 \ \Delta B\|$. □

The main result of this section establishes the equivalence of $\tau(P, B, \gamma)$ to the solution of the singular value minimization problem

$$\xi(P, B, \gamma) = \inf_{\lambda \in \mathbb{C}} \sigma_{\min} \left[\frac{P(\lambda)}{p_\gamma(|\lambda|)} \} \quad B \right], \tag{5.2}$$

where $p_\gamma(x)$ is as defined by (1.11). The next lemma eliminates the possibility that $\xi(P, B, \gamma)$ is attained at ∞.

Lemma 33. *Under the assumption that the leading coefficient of (5.1) is non-singular and remains nonsingular under perturbations with norm less than or equal to $\gamma_k \xi(P, B, \gamma)$, the inequality*

$$\xi(P, B, \gamma) < \lim_{\lambda \to \infty} \sigma_{\min} \left[\frac{P(\lambda)}{p_\gamma(|\lambda|)} \quad B \right].$$

holds.

Proof. When $\gamma_k = 0$, the result immediately follows. When $\gamma_k > 0$, we have

$$\sigma_{\min} \left[\frac{K_k}{\gamma_k} \quad B \right] = \lim_{\lambda \to \infty} \sigma_{\min} \left[\frac{P(\lambda)}{p_\gamma(|\lambda|)} \quad B \right].$$

Suppose that $\xi(P, B, \gamma)$ is attained at ∞ and therefore there exist $u_1, v \in \mathbb{C}^n$ and $u_2 \in \mathbb{C}^m$ such that

$$\begin{bmatrix} \left(\frac{K_k}{\gamma_k}\right)^* \\ B^* \end{bmatrix} v = \xi(P, B, \gamma) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where $[u_1^T \ u_2^T]^T$ and v have unit length. Multiplying the upper blocks by γ_k, the right-hand side by $v^* v$ and collecting all terms on the left yield

$$\begin{bmatrix} K_k^* - \gamma_k \xi(P, B, \gamma) u_1 v^* \\ B^* - \xi(P, B, \gamma) u_2 v^* \end{bmatrix} v = 0.$$

Consequently a perturbation to the leading coefficient with norm at most $\gamma_k \xi(P, B, \gamma)$ yields the singular matrix $K_k - \gamma_k \xi(P, B, \gamma) v u_1^*$, which contradicts the non-singularity assumption. □

Theorem 34. *With the assumptions of Lemma 33 for the system (5.1), the equality $\tau(P,B,\gamma) = \xi(P,B,\gamma)$ holds for τ defined in (1.19) both in the spectral norm and in the Frobenius norm.*

Proof. First we assume that $\tau(P,B,\gamma)$ in (1.19) is defined in the spectral norm and show that $\xi(P,B,\gamma) \le \tau(P,B,\gamma)$. From Lemma 32, there exists $\Delta P(\lambda) = \sum_{j=0}^{k} \gamma_j \lambda^j \Delta K_j$ such that

$$\tau(P,B,\gamma) = \|\Delta K_k \ \ \ldots \ \ \Delta K_0 \ \ \Delta B\|$$

and for some $\tilde{\lambda}$ the matrix $[(P+\Delta P)(\tilde{\lambda}) \ \ B+\Delta B]$ is rank-deficient, that is

$$\begin{bmatrix} ((P+\Delta P)(\tilde{\lambda}))^* \\ B^* + \Delta B^* \end{bmatrix} v = 0$$

for some unit $v \in \mathbb{C}^n$. We collect the perturbations on the right and divide the upper blocks by $p_\gamma(|\tilde{\lambda}|)$ to obtain

$$\begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ B^* \end{bmatrix} v = \begin{bmatrix} \left(-\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ -\Delta B^* \end{bmatrix} v.$$

Therefore

$$\begin{aligned} \xi(P,B,\gamma) \le \ \ & \sigma_{\min}\left[\frac{P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)} \ \ B\right] = \sigma_{\min}\begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ B^* \end{bmatrix} \le \left\| \begin{bmatrix} \left(\frac{P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ B^* \end{bmatrix} v \right\| \\ = \ \ & \left\| \begin{bmatrix} \left(\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ \Delta B^* \end{bmatrix} v \right\| \le \left\| \begin{bmatrix} \left(\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)}\right)^* \\ \Delta B^* \end{bmatrix} \right\| = \left\| \left[\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)} \ \ \Delta B\right] \right\|. \end{aligned}$$

Moreover,

$$\left[\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)} \ \ \Delta B\right] = [\Delta K_k \ \ \ldots \ \ \Delta K_0 \ \ \Delta B] \begin{bmatrix} \frac{\gamma_k \tilde{\lambda}^k I}{p_\gamma(|\tilde{\lambda}|)} & 0 \\ \vdots & \vdots \\ \frac{\gamma_1 \tilde{\lambda} I}{p_\gamma(|\tilde{\lambda}|)} & 0 \\ \frac{\gamma_0 I}{p_\gamma(|\tilde{\lambda}|)} & 0 \\ 0 & I \end{bmatrix}$$

where the spectral norm of the rightmost matrix is one. It follows from the Cauchy-Schwarz inequality that

$$\xi(P,B,\gamma) \le \left\| \left[\frac{\Delta P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)} \ \ \Delta B\right] \right\| \le \|[\Delta K_k \ \ \ldots \ \ \Delta K_0 \ \ \Delta B]\| = \tau(P,B,\gamma).$$

For the reverse inequality, still using the spectral norm, we have from Lemma 33 that for some φ,

$$\xi(P,B,\gamma) = \sigma_{\min}\begin{bmatrix}\dfrac{P(\varphi)}{p_\gamma(|\varphi|)} & B\end{bmatrix} = \sigma_{\min}\begin{bmatrix}\left(\frac{P(\varphi)}{p_\gamma(|\varphi|)}\right)^* \\ B^*\end{bmatrix}$$

or equivalently

$$\begin{bmatrix}\frac{(P(\varphi))^*}{p_\gamma(|\varphi|)} \\ B^*\end{bmatrix} v = \xi(P,B,\gamma)\begin{bmatrix}u_1 \\ u_2\end{bmatrix},$$

where $v, u_1 \in \mathbb{C}^n$, $u_2 \in \mathbb{C}^m$ and the vectors v and $[u_1^T\ u_2^T]^T$ have unit length. We multiply the right-hand side by v^*v, the upper blocks by $p_\gamma(|\varphi|)$ and collect all terms on the left to obtain

$$\begin{bmatrix}(P(\varphi))^* - p_\gamma(|\varphi|)\xi(P,B,\gamma)u_1v^* \\ B^* - \xi(P,B,\gamma)u_2v^*\end{bmatrix} v = 0.$$

In other words, the matrix

$$[P(\varphi) - p_\gamma(|\varphi|)\xi(P,B,\gamma)vu_1^* \quad B - \xi(P,B,\gamma)vu_2^*]$$

is rank-deficient. If we set $\Delta K_j = \frac{-\gamma_j\bar{\varphi}^j\xi(P,B,\gamma)vu_1^*}{p_\gamma(|\varphi|)}$ and $\Delta B = -\xi(P,B,\gamma)vu_2^*$ and define $\Delta P(\lambda) = \sum_{j=0}^m \gamma_j\lambda^j\Delta K_j$, then noting that

$$\Delta P(\varphi) = \sum_{j=0}^m \gamma_j\varphi^j\Delta K_j = -p_\gamma(|\varphi|)\xi(P,B,\gamma)vu_1^*$$

we see that

$$[(P+\Delta P)(\lambda)\ \ B+\Delta B]$$

is rank deficient at $\lambda = \varphi$. The norm of the perturbations satisfies

$$\|\Delta K_k\ \ \ldots\ \ \Delta K_0\ \ \Delta B\| = \xi(P,B,\gamma)\left\|\gamma_k\bar{\varphi}^k\frac{vu_1^*}{p_\gamma(|\varphi|)}\ \ \ldots\ \ \gamma_0\frac{vu_1^*}{p_\gamma(|\varphi|)}\ \ vu_2^*\right\| \le \xi(P,B,\gamma).$$

Therefore $\tau(P,B,\gamma) \le \|\Delta K_k\ \ \ldots\ \ \Delta K_0\ \ \Delta B\| \le \xi(P,B,\gamma)$ as desired.

For the claim about the equality when $\tau(P,B,\gamma)$ is defined in the Frobenius norm, to show that $\xi(P,B,\gamma) \le \tau(P,B,\gamma)$ the proof in the first part applies noting that

$$\xi(P,B,\gamma) \le \|\Delta K_k\ \ \ldots\ \ \Delta K_0\ \ \Delta B\|_2 \le \|\Delta K_k\ \ \ldots\ \ \Delta K_0\ \ \Delta B\|_F = \tau(P,B,\gamma).$$

The second part, to show that $\tau(P,B,\gamma) \le \xi(P,B,\gamma)$, applies without modification. □

The second part of Theorem 34 explicitly constructed the closest uncontrollable system, which we state in the next corollary.

Corollary 35. *Let $\xi(P, B, \gamma)$ be attained at λ_* and the vectors $[u_1^T \; u_2^T]^T$ and v be the unit right and left singular vectors corresponding to*

$$\sigma_{\min}\left[\begin{array}{cc} \dfrac{P(\lambda_*)}{p_\gamma(|\lambda_*|)} & B \end{array}\right],$$

respectively, where $u_1, v \in \mathbb{C}^n$ and $u_2 \in \mathbb{C}^m$. A closest uncontrollable tuple is $(K_k + \gamma_k \Delta K_k, \ldots, K_0 + \gamma_0 \Delta K_0, B + \Delta B)$, where

$$\Delta K_j = \frac{-\gamma_j \bar{\lambda}_*^j \xi(P, B, \gamma) v u_1^*}{p_\gamma(|\lambda_*|)}, \quad j = 0, \ldots, k$$

and

$$\Delta B = -\xi(P, B, \gamma) v u_2^*.$$

5.2 A practical algorithm exploiting the singular value characterization

In Theorem 34 we established the equality

$$\tau(P, B, \gamma) = \xi(P, B, \gamma) = \inf_{r \geq 0, \theta \in [0, 2\pi)} f(r, \theta)$$

where

$$f(r, \theta) = \sigma_{\min}\left[\begin{array}{cc} \dfrac{P(re^{i\theta})}{p_\gamma(r)} & B \end{array}\right].$$

In this section we present a trisection algorithm to minimize the function $f(r, \theta)$ in polar coordinates. Let δ_1 and δ_2 trisect the interval $[L, U]$ containing the distance to uncontrollability (see Figure 4.1). Set

$$\delta = \delta_1, \quad \eta = \frac{2}{k} \arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right).$$

We say the angle η subtends all of the components of the δ-level set of f when none of the components has a pair of points whose angles differ by more than η. At each iteration our aim is to ensure that either the δ-level set of f is not empty or the angle η subtends all of the components of the δ-level set of f. By the definition of $\xi(P, B, \gamma)$, when the δ-level set is not empty

$$\delta = \delta_1 \geq \xi(P, B, \gamma), \tag{5.3}$$

and when η subtends all of the components of the δ-level set we will see below that

$$\xi(P, B, \gamma) > \delta_2 \tag{5.4}$$

because of the choice of η and δ. The trisection algorithm starts with the trivial upper bound $U = \sigma_{\min}[K_k/\gamma_k \;\; B]$ (or when $\gamma_k = 0$, for some $\tilde{r}$ and $\tilde{\theta}$, $U = f(\tilde{r}, \tilde{\theta})$) and the lower bound $L = 0$. At each iteration we either update the upper bound to δ_1 if the inequality (5.3) is verified or the lower bound to δ_2 if the inequality (5.4) is verified.

First we need to be equipped with a technique that checks for a given δ and θ whether there exists an r satisfying

$$f(r, \theta) = \delta, \tag{5.5}$$

that is whether the line with slope θ passing through the origin, say $\mathcal{L}(\theta)$, intersects the δ-level set of f. Our first result in this section shows how this can be achieved by solving a $*$-even polynomial eigenvalue problem of double size and of double degree.

Theorem 36. *Given $\theta \in [0, 2\pi)$ and a positive real number δ, the matrix $\left[\frac{P(re^{i\theta})}{p_\gamma(r)} \;\; B\right]$ has δ as a singular value if and only if the matrix polynomial of double size $Q(\lambda, \theta, \delta) = \sum_{j=0}^{2k} \lambda^j Q_j(\theta, \delta)$ has the imaginary eigenvalue ri where*

$$Q_0(\theta, \delta) = \begin{bmatrix} -\delta\gamma_0^2 I & K_0^* \\ K_0 & BB^*/\delta - \delta I \end{bmatrix},$$

and, when l is odd,

$$\begin{aligned} Q_l(\theta, \delta) &= \begin{bmatrix} 0 & (-1)^{(l+1)/2} i K_l^* e^{-il\theta} \\ (-1)^{(l+1)/2} i K_l e^{il\theta} & 0 \end{bmatrix} && 1 \le l \le k, \\ Q_l(\theta, \delta) &= 0 && k+1 \le l < 2k, \end{aligned}$$

and, when l is even,

$$\begin{aligned} Q_l(\theta, \delta) &= \begin{bmatrix} (-1)^{l/2+1}\delta\gamma_{l/2}^2 I & (-1)^{l/2} K_l^* e^{-il\theta} \\ (-1)^{l/2} K_l e^{il\theta} & 0 \end{bmatrix} && 1 \le l \le k, \\ Q_l(\theta, \delta) &= \begin{bmatrix} (-1)^{l/2+1}\delta\gamma_{l/2}^2 I & 0 \\ 0 & 0 \end{bmatrix} && k+1 \le l \le 2k. \end{aligned}$$

Proof. The matrix $\left[\frac{P(re^{i\theta})}{p_\gamma(r)} \;\; B\right]$ has δ as a singular value if and only if both of the equations

$$\left[\frac{P(re^{i\theta})}{p_\gamma(r)} \;\; B\right] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \delta u$$

and

$$\begin{bmatrix} \left(\frac{P(re^{i\theta})}{p_\gamma(r)}\right)^* \\ B^* \end{bmatrix} u = \delta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

are satisfied. From the bottom block of the second equation we have $v_2 = B^*u/\delta$. By eliminating v_2 from the other equations, we obtain

$$\begin{bmatrix} -\delta I & \left(\frac{P(re^{i\theta})}{p_\gamma(r)}\right)^* \\ \frac{P(re^{i\theta})}{p_\gamma(r)} & BB^*/\delta - \delta I \end{bmatrix} \begin{bmatrix} v_1 \\ u \end{bmatrix} =$$

$$\begin{bmatrix} -\delta p_\gamma(r) I & \left(P(re^{i\theta})\right)^* \\ P(re^{i\theta}) & BB^*/\delta - \delta I \end{bmatrix} \begin{bmatrix} v_1/p_\gamma(r) \\ u \end{bmatrix} = \sum_{j=0}^{2k} (ri)^j Q_j(\theta, \delta) \begin{bmatrix} v_1/p_\gamma(r) \\ u \end{bmatrix} = 0.$$

Therefore ri is an eigenvalue of $Q(\lambda, \theta, \delta)$. □

Suppose $\delta \leq \lim_{\lambda\to\infty} \sigma_{\min}\left[\frac{P(\lambda)}{p_\gamma(|\lambda|)} \;\; B\right]$. To establish the existence of an r satisfying (5.5), it is sufficient that the polynomial $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue. When $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue $r'i$, $f(r', \theta) \leq \delta$. Since $\delta \leq f(r, \theta)$ in the limit as $r \to \infty$, by the continuity of f we deduce $f(\hat{r}, \theta) = \delta$ for some $\hat{r} \geq r'$.

For our trisection algorithm it suffices to check whether any of the lines $\mathcal{L}(0)$, $\mathcal{L}(\eta)$, $\mathcal{L}(2\eta)$, …, $\mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersects the δ-level set of f. When there is an intersection point the δ-level set is not empty; otherwise the angle η subtends all of the components. The only part of the algorithm that is not clarified so far is how we deduce a lower bound on $\xi(P, B, \gamma)$ when η subtends all of the components, in particular the relation between δ_2 in (5.4) and the pair δ and η. For the next theorem addressing these issues, let (r_*, θ_*) be a point where $\xi(P, B, \gamma)$ is attained. We assume the existence of a constant c known *a priori* satisfying

$$c \geq \max_{0\leq j\leq k} \frac{|r_*|^j}{p_\gamma(r_*)} = \max\left(\frac{1}{p_\gamma(r_*)}, \frac{|r_*|^k}{p_\gamma(r_*)}\right). \tag{5.6}$$

Finding a constant c may be tedious in some special cases. However, when both γ_k and γ_0 are nonzero we can set $c = \frac{1}{\min(\gamma_0, \gamma_k)}$. We furthermore use the notation $K_{\max} = \max_{1\leq j\leq k} \|K_j\|$.

Theorem 37. *Let*

$$\lim_{\lambda\to\infty} \sigma_{\min}\left[\frac{P(\lambda)}{p_\gamma(|\lambda|)} \;\; B\right] \geq \delta > \tau(P, B, \gamma).$$

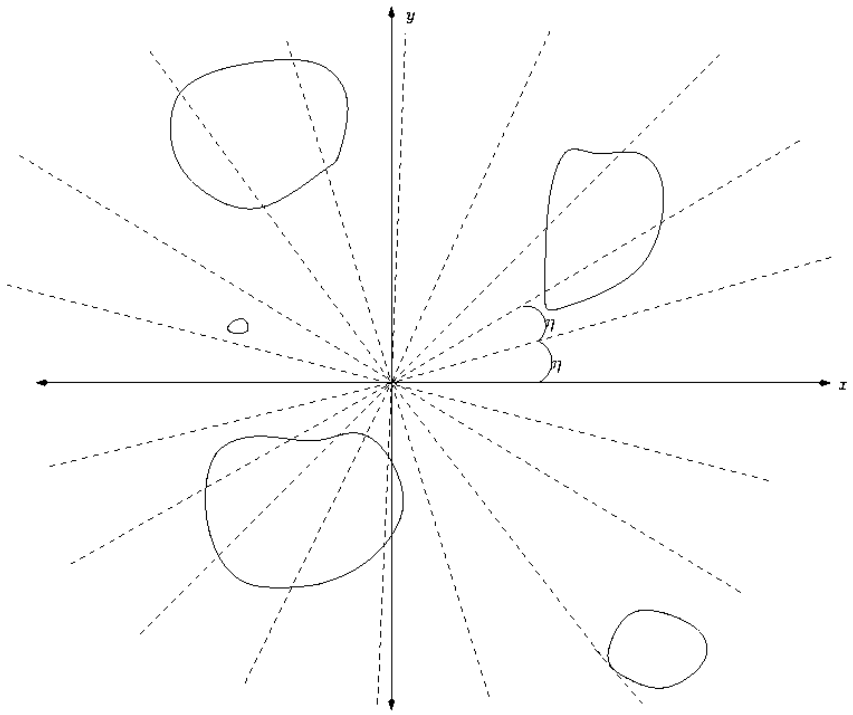

Figure 5.1: To verify which one of (5.3) and (5.4) holds we check the intersection points of the δ-level set of f and the set of lines with slopes multiples of η ranging from 0 to π. The closed curves are the δ-level curves.

Given any $\eta \in \left[0, \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta-\tau}{ckK_{\max}}\right)^2\right)\right]$, *there exist* r_1 *and* r_2 *(depending on* η*) such that*

$$\sigma_{\min}\left[\frac{P(r_1 e^{i(\theta_*+\eta)})}{p_\gamma(r_1)} \;\; B\right] = \delta \quad \text{and} \quad \sigma_{\min}\left[\frac{P(r_2 e^{i(\theta_*-\eta)})}{p_\gamma(r_2)} \;\; B\right] = \delta.$$

Proof. We prove only the first equality, since the proof of the second equality is similar. Assume that

$$\sigma_{\min}\left[\frac{P(re^{i(\theta_*+\eta)})}{p_\gamma(r)} \;\; B\right] > \delta \tag{5.7}$$

holds for all r for an η in the interval specified. Since the singular values of a matrix X are the eigenvalues of the symmetric matrix

$$\begin{bmatrix} 0 & X \\ X^* & 0 \end{bmatrix},$$

they are globally Lipschitz with constant 1 meaning that

$$\delta - \tau < \sigma_{\min}\left[\frac{P(r_* e^{i(\theta_*+\eta)})}{p_\gamma(r_*)} \;\; B\right] - \sigma_{\min}\left[\frac{P(r_* e^{i\theta_*})}{p_\gamma(r_*)} \;\; B\right] \leq$$
$$\left\|\left[\frac{P(r_* e^{i(\theta_*+\eta)})}{p_\gamma(r_*)} \;\; B\right] - \left[\frac{P(r_* e^{i\theta_*})}{p_\gamma(r_*)} \;\; B\right]\right\| = \left\|\frac{\sum_{j=1}^k r_*^j e^{ij\theta_*} K_j(e^{ij\eta}-1)}{p_\gamma(r_*)}\right\|.$$

Notice that $\eta \leq \pi/k$, implying that $\cos k\eta \leq \cos j\eta$ for $j = 0, \ldots, k$. Therefore

$$kcK_{\max}\sqrt{2-2\cos k\eta} \geq \sum_{j=1}^k c\|K_j\sqrt{2-2\cos j\eta}\| \geq \left\|\frac{\sum_{j=1}^k r_*^j e^{ij\theta_*} K_j(e^{ij\eta}-1)}{p_\gamma(r_*)}\right\| > \delta - \tau$$

or

$$1 - \frac{1}{2}\left(\frac{\delta-\tau}{kcK_{\max}}\right)^2 > \cos k\eta.$$

Since the cos function is strictly decreasing in the interval $[0, \pi]$, we obtain the contradiction that

$$\eta > \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta-\tau}{kcK_{\max}}\right)^2\right).$$

Thus, (5.7) cannot hold, so there exists r_1' satisfying

$$\sigma_{\min}\left[\frac{P(r_1' e^{i(\theta_*+\eta)})}{p_\gamma(r_1')} \;\; B\right] \leq \delta.$$

The first equality of the theorem must therefore hold for some $r_1 \geq r_1'$ because of the continuity of $f(r, \theta_* + \eta)$ with respect to r and the fact that $\lim_{r\to\infty} f(r, \theta_* + \eta) \geq \delta$. □

As we have already indicated in (5.3), we first set $\delta = \delta_1$. The assignment

$$\eta = \frac{2}{k} \arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right) \tag{5.8}$$

leads us to the lower bound (5.4) in the case that none of the lines $\mathcal{L}(0)$, $\mathcal{L}(\eta)$, $\mathcal{L}(2\eta)$, $\ldots$, $\mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersects the δ-level set of f, which we can see as follows. According to Theorem 37 for all θ in the interval

$$\left[\theta_* - \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \tau}{ckK_{\max}}\right)^2\right), \theta_* + \frac{1}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \tau}{ckK_{\max}}\right)^2\right)\right], \tag{5.9}$$

the line $\mathcal{L}(\theta)$ intersects the δ-level set of f. When none of the lines $\mathcal{L}(0), \mathcal{L}(\eta), \mathcal{L}(2\eta)$, $\ldots \mathcal{L}(\lfloor \frac{\pi}{\eta} \rfloor \eta)$ intersects the δ-level set of f, it follows that η must be greater than the length of the interval in (5.9), that is

$$\eta = \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right) > \frac{2}{k}\arccos\left(1 - \frac{1}{2}\left(\frac{\delta - \tau}{ckK_{\max}}\right)^2\right).$$

From this inequality it is straightforward to deduce the lower bound (5.4).

We summarize the algorithm below (Algorithm 8). The standard way to solve a polynomial eigenvalue problem of size $2n$ and degree $2k$ is to reduce it to an equivalent generalized eigenvalue problem $\mathcal{H} + \lambda\mathcal{N}$ of size $4nk$ and use a generalized eigenvalue solver with cubic complexity. At each iteration the algorithm below requires the solution of the eigenvalue problems $Q(\lambda, 0, \delta), Q(\lambda, \eta, \delta), \ldots, Q(\lambda, \lfloor \frac{\pi}{\eta} \rfloor \eta, \delta)$ each typically at a cost of $O(n^3k^3)$. The overall complexity of an iteration is

$$O\left(\frac{n^3k^4}{\arccos\left(1 - \frac{1}{2}\left(\frac{\delta_1 - \delta_2}{ckK_{\max}}\right)^2\right)}\right). \tag{5.10}$$

It is apparent that the initial iterations for which $\delta_1 - \delta_2$ is relatively large are cheaper, while the last iteration for which $\delta_1 - \delta_2 \approx tol/2$ is the most expensive. Note also that the choice of c affects the number of lines and therefore the running time. It is not desirable to set it very large.

For reliability the method for solving the $*$-even polynomial eigenvalue problem $Q(\lambda, \theta, \delta)$ is especially important. To avoid using tolerances to decide whether a computed eigenvalue can be accepted as imaginary, a $*$-even polynomial eigenvalue solver respecting the symmetry of the spectrum (as discussed in §A.3) must be used.

Algorithm 8 Trisection algorithm for the higher-order distance to uncontrollability

Call:	$[L, U] \leftarrow$ `HODU`(P,B,γ,tol,c).
Input:	$P \in \mathbb{C}^{k\times n\times n}$ (the matrix polynomial), $B \in \mathbb{C}^{n\times m}$, $\gamma \in \mathbb{R}^k$ (non-negative scaling factors, not all zero), tol (desired tolerance), c (a positive real number satisfying (5.6)).
Output:	L, U with $L < U$, $U - L \leq tol$. The interval $[L, U]$ contains the higher-order distance to uncontrollability.

Initially for some $\tilde{\lambda}$ set

$$U \leftarrow \sigma_{\min}\begin{bmatrix}\frac{K_k}{\gamma_k} & B\end{bmatrix} \quad \text{if } \gamma_k > 0$$
$$U \leftarrow \sigma_{\min}\begin{bmatrix}\frac{P(\tilde{\lambda})}{p_\gamma(|\tilde{\lambda}|)} & B\end{bmatrix} \quad \text{if } \gamma_k = 0$$

and $L \leftarrow 0$.
while $U - L > tol$ **do**
  Set $\delta_1 \leftarrow L + 2(U - L)/3$ and $\delta_2 \leftarrow L + (U - L)/3$.
  Set $\delta \leftarrow \delta_1$ and η as defined in (5.8)
  Set $Intersection \leftarrow FALSE$.
  for $\theta = 0$ to π in increments of η **do**
    Compute the eigenvalues of $Q(\lambda, \theta, \delta)$.
    if $Q(\lambda, \theta, \delta)$ has an imaginary eigenvalue **then**
      % An intersection point is detected
      Update the upper bound, $U \leftarrow \delta_1$.
      $Intersection \leftarrow TRUE$.
      Break. (Leave the for loop.)
    end if
  end for
  if $\neg Intersection$ **then**
    % No intersection point is detected
    Update the lower bound, $L \leftarrow \delta_2$.
  end if
end while
Return $[L, U]$.

iteration	total running time	Interval $[L, U]$
1	0.400	[0.000,0.667]
2	1.680	[0.222,0.667]
3	2.510	[0.222,0.519]
4	5.369	[0.321,0.519]
5	9.670	[0.387,0.519]
6	16.110	[0.431,0.519]
7	20.140	[0.431,0.489]
8	34.580	[0.450,0.489]
9	56.770	[0.463,0.489]
10	70.470	[0.463,0.481]
11	118.40	[0.469,0.481]
12	190.93	[0.473,0.481]

Table 5.1: Total running time of the trisection algorithm after each iteration on a Toeplitz matrix and vector pair.

5.3 Numerical examples

5.3.1 The special case of first-order systems

Even though it is much slower than the methods in [31, 13] and the method based on real eigenvalue extraction techniques in the previous chapter, Algorithm 8 can be applied to estimate the first-order distance to uncontrollability with $k = 1$, $K_1 = I$ and $\gamma = [0\ 1]$ so that perturbations to $K_1 = I$ are not allowed. It is well known that in this case the distance to uncontrollability is attained at a point λ_* with $|\lambda_*| = c \leq 2(\|K_0\| + \|B\|)$ [16]. We choose K_0 as the Toeplitz matrix

$$\begin{bmatrix} 1 & 3 & 0 & 0 \\ -2 & 1 & 3 & 0 \\ 0 & -2 & 1 & 3 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

and $B = [2\ 2\ 2\ 2]^T$. When we require an interval of length 10^{-2} or less, Algorithm 8 returns $[0.473, 0.481]$ in 12 iterations which contains the distance to uncontrollability 0.477. Table 5.1 lists the cumulative running time after each iteration in seconds. Overall we observe that reaching one-digit accuracy is considerably cheaper than two-digit accuracy. When we allow perturbations to the leading coefficient by setting $\gamma = [1\ 1]$, there is a closer uncontrollable system at a distance of $\tau(P, B, \gamma) \leq 0.145$ which is the upper bound returned by Algorithm 8.

5.3.2 A quadratic brake model

In [27] the vibrations of a drum brake system are modeled by the quadratic equation

$$Mx^{(2)}(t) + K(\mu)x(t) = f(t) \tag{5.11}$$

with the mass and stiffness matrices

$$M = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}, \quad K(\mu) = g \begin{bmatrix} (\sin\gamma + \mu\cos\gamma)\sin\gamma & -\mu - (\sin\gamma + \mu\cos\gamma)\cos\gamma \\ (\mu\sin\gamma - \cos\gamma)\sin\gamma & 1 + (-\mu\sin\gamma + \cos\gamma)\cos\gamma \end{bmatrix}.$$

Suppose that the force on the brake system has just the vertical component determined by the input

$$f(t) = \begin{bmatrix} f_x(t) \\ f_y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

For the parameters $m = 5$, $g = 1$ and $\gamma = \frac{\pi}{100}$, we consider two cases. First by setting $\gamma = [1\ 0\ 1]$, we impose equal importance on the perturbations to the mass and stiffness matrices. Notice that for small μ and γ, the system is close to being uncontrollable. In the second column in Table 5.2 the intervals of length 10^{-2} or less containing the distance to uncontrollability returned by Algorithm 8 are provided for various values of μ. The algorithm iterates 16 times to reach two-digit accuracy. Secondly we assign scaling to the perturbations proportional to the norms of the mass and stiffness matrices, that is $\gamma = [\|M\|\ 0\ \|K\|]$. The intervals returned by Algorithm 8 for this second case are given in the rightmost column in Table 5.2. As expected, the distance to uncontrollability again increases with respect to μ. The system (5.11) is closer to being uncontrollable in a relative sense than in an absolute sense.

If we allow perturbations to all coefficients with equal scaling (*i.e.* $\gamma = [1\ 1\ 1]$), usually the first-order distance to uncontrollability of the linearized system is considerably smaller than the higher-order distance to uncontrollability $\tau(P, B, \gamma)$. This is because the perturbations in the definition of the first-order distance to uncontrollability are not constrained, so they don't respect the structure of the linearization. For instance, for the drum brake system with $\gamma = [1\ 1\ 1]$ and $\mu = 0.1$, $\tau(P, B, \gamma) \in [0.097, 0.105]$ (note that this is same as the entry for $\mu = 0.1$ in the second column in Table 5.2; up to two-digit accuracy it does not make any difference whether we allow perturbations to the coefficient K_1 or not) whereas the standard unstructured distance to uncontrollability of the embedding lies in the interval $[0.012, 0.020]$.

μ	**Interval $[L,U]$ (Absolute)**	**Interval $[L,U]$ (Relative)**
0.05	[0.051,0.059]	[0.038,0.046]
0.10	[0.097,0.105]	[0.071,0.079]
0.15	[0.140,0.148]	[0.104,0.112]
0.20	[0.184,0.191]	[0.137,0.145]
0.50	[0.418,0.426]	[0.325,0.333]
1	[0.676,0.684]	[0.574,0.581]
10	[0.990,0.997]	[0.984,0.991]
100	[0.993,1.000]	[0.987,0.994]
1000	[0.993,1.000]	[0.987,0.994]

Table 5.2: The intervals computed by the trisection algorithm for the brake system for various μ values in an absolute sense in the second column and in a relative sense in the third column.

size / order	**first-order**	**quadratic**	**cubic**
5	10.260 (10)	192.760 (12)	1237.540 (13)
10	83.550 (12)	1392.070 (11)	12485.740 (12)
15	271.560 (13)	6390.000 (14)	37324.310 (12)

Table 5.3: Running time of the trisection algorithm in seconds with respect to the size and the order of the systems with normally distributed coefficient matrices.

5.3.3 Running time with respect to the size and the order of the system

We run the trisection algorithm on systems with random coefficients of various size and order. To be precise, the entries of all of the coefficient matrices were chosen from a normal distribution with zero mean and variance one independently. Table 5.3 illustrates how the running time in seconds varies with respect to the size and the order of the system. In all of the examples intervals of length at most 10^{-2} containing the absolute distance to uncontrollability (γ is the vector of ones) were returned. The numbers in parentheses correspond to the number of trisection iterations needed. The variation in the running time with respect to the size and the order is consistent with the complexity suggested by (5.10).

Chapter 6

Software and Open Problems

We conclude this thesis by listing all the available software implemented to compute the robust stability and controllability measures. We also describe some related open problems.

6.1 Software

All of the algorithms in this thesis are implemented and tested in MATLAB. For these particular algorithms, programming in a traditional high-level language such as C or FORTRAN does not provide a significant improvement in the running time, as the computations are usually dominated by the solutions of eigenvalue problems that are performed in MATLAB by calling LAPACK or ARPACK routines.

The MATLAB routines for robust stability and controllability are available on the web site [58] as a *tar* and a *zip* file. When the *tar* or *zip* file is extracted, the directory `software_rob_sc` is created. In this directory the routines are collected in six subdirectories.

- The subdirectory `auxiliary` contains common routines that are called by the main routines frequently for purposes such as error handling and the solution of polynomial eigenvalue problems. The user does not need to know the details of the routines in this subdirectory.

- The subdirectory `visualization` contains routines to view the functions (or their level sets) that are optimized in this thesis; these are listed in Table 6.1. The subdirectory `auxiliary` must be added to the MATLAB path before running the visualization routines. Additionally for the routines `plot_kreiss_constant_cont` and `plot_kreiss_constant_disc`, the subdirectories

`plot_cdi`	Plots the function $g(\omega) = \sigma_{\min}[A - \omega i I]$ in a given real interval.
`plot_ddi`	Plots the function $g(\theta) = \sigma_{\min}[A - e^{i\theta} I]$ in a given subinterval of $[0, 2\pi)$.
`plot_kreiss_constant_cont`	Plots the ratio α_ϵ/ϵ for ϵ in a given real interval on a log 10 scale.
`plot_kreiss_constant_disc`	Plots the ratio $(\rho_\epsilon - 1)/\epsilon$ for ϵ in a given real interval on a log 10 scale.
`plot_num_rad`	Plots the function $f(\theta) = \lambda_{\max}(H(Ae^{i\theta}))$ in a given subinterval of $[0, 2\pi)$.
`plot_poly_cdi`	Plots the function $h(\omega) = \sigma_{\min}[P(\omega i)]/p_\gamma(\lvert\omega\rvert)$ in a given real interval and for a given scaling γ.
`plot_poly_ddi`	Plots the function $h(\theta) = \sigma_{\min}[P(e^{i\theta})]/\lVert\gamma\rVert$ in a given subinterval of $[0, 2\pi)$ and for a given scaling.
`poly_ps`	Draws the ϵ-pseudospectra of a given matrix polynomial for a given scaling γ and various ϵ.
`rect_poly_ps`	Draws the δ-level sets of the function $\sigma_{\min}[P(z)/p_\gamma(\lvert z\rvert) \quad B]$ in the complex plane for a given matrix polynomial P, a matrix B, the scaling γ and various δ.

Table 6.1: Visualization routines

`fo_robust_stability/pseudo_abscissa`

and

`fo_robust_stability` /`pseudo_radius`

must be added to the MATLAB path. To draw the level sets, the functions `poly_ps` and `rect_poly_ps` perform radial searches in various directions defined in Theorem 21 and Theorem 36.

- The routines for the measures for the robust stability of first-order systems can be found in the subdirectory `fo_robust_stability`. This subdirectory contains five subdirectories, one for each of the pseudospectral abscissa, pseudospectral radius, numerical radius, continuous distance to instability and discrete distance to instability. The main routines to compute these measures are listed in Table 6.2.

`pspa`	Computes the ϵ-pseudospectral abscissa of a matrix. This routine is included in the subdirectory `pseudo_abscissa`.
`pspr`	Computes the ϵ-pseudospectral radius of a matrix. The routine is in the subdirectory `pseudo_radius`.
`cdi`	Computes the continuous distance to instability of a matrix. The routine is in the subdirectory `distance_inst_cont`.
`ddi`	Computes the discrete distance to instability of a matrix. The routine is in the subdirectory `distance_inst_disc`.
`numr`	Computes the numerical radius of a matrix. The routine is in the subdirectory `numerical_radius`.

Table 6.2: Routines for the computation of the first-order robust stability measures.

- The routines for robust stability of higher-order systems are in the subdirectory `ho_robust_stability`. This subdirectory contains subdirectories for the computation of the pseudospectral abscissa, pseudospectral radius, continuous distance to instability and discrete distance to instability of a matrix polynomial. The main routines are listed in Table 6.3.

- The routines for the first-order distance to uncontrollability are included in the subdirectory `fo_robust_controllability`. The main routine that should be called for the first-order distance to uncontrollability is `dist_uncont_hybrid`. By setting the input parameters `options.method` and `options.eigmethod` appropriately, various algorithms described in this thesis can be run. The parameter `options.method` is used to determine whether the trisection algorithm for high precision or low precision or the BFGS algorithm will be used. Based on the value of parameter `options.eigmethod`, either the real eigenvalue extraction technique is used or otherwise `eig` of MATLAB will be called. All of the possible combination of parameter values and the corresponding algorithms that will be executed are given in Table 6.4. For the algorithm for low precision, regardless of the value of `options.eigmethod`, `eig` of MATLAB will be called as the real eigenvalue extraction technique is not applicable to this case. The default values for these parameters are `options.method`= 1 and `options.eigmethod`= 1 which means that the BFGS algorithm with the

`poly_pspa`	Computes the ϵ-pseudospectral abscissa of a matrix polynomial. The routine is in the subdirectory `pseudo_abscissa`.
`poly_pspr`	Computes the ϵ-pseudospectral radius of a matrix polynomial. The routine is in the subdirectory `pseudo_radius`.
`poly_cdi`	Computes the continuous distance to instability of a matrix polynomial. The routine is in the subdirectory `distance_inst_cont`.
`poly_ddi`	Computes the discrete distance to instability of a matrix polynomial. The routine is in the subdirectory `distance_inst_disc`.

Table 6.3: Routines for the computation of the higher-order robust stability measures.

	`method`= 0	`method`= 1	`method`= 2
`eigmethod`= 0	The trisection algorithm for high precision (with the new verification scheme) in §4.3 without the divide-and-conquer real eigenvalue extraction	The BFGS algorithm described in §4.3.4 (using the new verification scheme) without the divide-and-conquer real eigenvalue extraction technique	The trisection algorithm for low precision in §4.2
`eigmethod`= 1	The trisection algorithm for high precision in §4.3 with the divide-and-conquer real eigenvalue extraction	The BFGS algorithm described in §4.3.4 with the divide-and-conquer real eigenvalue extraction technique	The trisection algorithm for low precision in §4.2

Table 6.4: The first-order distance to uncontrollability algorithms that are implemented.

real eigenvalue extraction technique will be invoked if these parameters are not supplied.

- The routines for the higher-order distance to uncontrollability are in the subdirectory `ho_robust_controllability`. The main routine that implements Algorithm 8 is `poly_dist_uncont`.

Note that the current release of the codes does not use structure-preserving eigenvalue solvers and they can be run on any platform where MATLAB is installed except that the routine to compute the first-order distance to uncontrollability with the real eigenvalue extraction can only be run under Linux.[1] We

[1]When the routine `dist_uncont_hybrid` is called by setting `options.eigmethod`=1, the

hope to provide the routines using the structure-preserving eigenvalue solvers in the future.

6.2 Open problems

6.2.1 Large scale computation of robust stability measures

The numerical examples at the end of Chapter 2 and Chapter 3 show that the algorithms for the robust stability measures are feasible only for small to medium scale problems. All of the algorithms require extraction of the imaginary or unit eigenvalues of structured problems. We do not take advantage of the fact that we need only the eigenvalues on the imaginary axis or on the unit circle by using the QR algorithm or the structure-preserving methods in Appendix A. The distance of a given matrix A to the closest singular matrix is equal to $\sigma_{\min}(A)$ and would ideally be computed by an iterative method such as Arnoldi especially when A is large. On the other hand, in Chapter 4 we presented a real eigenvalue extraction technique based on iterative eigenvalue solvers which reduced the overall cost of the algorithm for the distance to uncontrollability from $O(n^6)$ to $O(n^4)$ on average. The real eigenvalue extraction can be modified for imaginary or unit eigenvalue extraction for efficient computation as long as the matrices whose imaginary or unit eigenvalues we seek can be inverted, shifted and multiplied onto a vector efficiently. The applicability of the real eigenvalue extraction for the computation of the robust stability measures of large scale matrices is still under investigation.

6.2.2 Kreiss constants

We presented efficient procedures for the computation of the ϵ-pseudospectral abscissa and the ϵ-pseudospectral radius of a medium size matrix. One of the major reasons why we are interested in the computation of the either one of these measures is to obtain the corresponding Kreiss constant defined by (2.22) for continuous systems and (3.37) for discrete systems. In the specific examples that we focused on, we usually observed that the Kreiss constants are attained at ϵ values slightly larger than the distance to instability. But in general it is difficult to guess *a priori* which ϵ value is most relevant for the transient peak.

algorithms for the first-order distance to uncontrollability with the real eigenvalue extraction technique will be invoked, requiring a Sylvester equation solver. In the current release a mex file to solve Sylvester equations is included only for Linux. The Sylvester equation solvers for other platforms will be made available in the next release.

Therefore it is desirable to design an algorithm for the efficient computation of the Kreiss constants. Indeed the alternative characterizations

$$\mathcal{K}_c(A) = \sup_{\text{Re } z>0} \frac{\text{Re } z}{\sigma_{\min}(A - zI)}$$

and

$$\mathcal{K}_d(A) = \sup_{|z|>1} \frac{|z| - 1}{\sigma_{\min}(A - zI)}$$

indicate the similarity of these problems to the distance to uncontrollability. It may be possible to extend the algorithms for the distance to uncontrollability to compute Kreiss constants. But as the algorithms for the distance to uncontrollability are expensive, we ideally would hope for more efficient techniques.

6.2.3 Computation of pseudospectra

At the moment the tool that is widely used for the computation of pseudospectra is the MATLAB toolbox *EigTool* [73], which benefits from the ideas in [69]. The most important observation that facilitates the computation is that the pseudospectra of similar matrices are identical. If $A = QTQ^*$ is a Schur decompositon for the matrix A, we can compute $\sigma_{\min}(T - zI)$ for various z on a 2-D grid. Using an iterative solver, the minimum singular value of $T - zI$ can typically be retrieved in $O(n^2)$ time, so the overall cost is $O(n^3 + s_2 n^2)$ where s_2 is the number of grid points. If we are specifically interested in the ϵ-pseudospectra for a few ϵ values, an alternative way is to perform the radial searches in various directions (see (3.9) for the definition of the radial search and Corollary 12 for how to perform it efficiently) on a 1-D grid. A straightforward implementation using direct eigenvalue solvers would have computational complexity $O(s_1 n^3)$ with s_1 denoting the number of points on the 1-D grid. Since we are only interested in the imaginary eigenvalues of the Hamiltonian matrices used for the radial searches, it may be possible to take advantage of the real eigenvalue extraction techniques as discussed in §6.2.1. Especially when we need to retrieve the ϵ-pseudospectrum for a particular ϵ on a fine grid, the method based on radial searches may have advantages over the methods working on a 2-D grid.

Appendix A

Structured Eigenvalue Problems

The algorithms that are presented in this thesis require the extraction of the imaginary eigenvalues of Hamiltonian matrices and even-odd matrix polynomials and the unit eigenvalues of symplectic pencils and palindromic matrix polynomials. Below we review these structured eigenvalue problems briefly.

A.1 Hamiltonian eigenvalue problems

A complex matrix M of size $2n$ is called *Hamiltonian* if the product JM is Hermitian where

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}. \tag{A.1}$$

A complex pencil $M_1 - \lambda M_2$ of size $2n$ is called *Hamiltonian* if $M_1 J M_2^*$ is Hermitian. It is easy to verify that both the Hamiltonian matrices and the Hamiltonian pencils have the eigenvalue symmetry $(\lambda, -\bar{\lambda})$ unless λ is imaginary. Suppose that λ' is an eigenvalue of the Hamiltonian matrix M with the right eigenvector x, then

$$Mx = \lambda' x \iff JMx = \lambda' Jx \iff x^* JM = -\bar{\lambda}' x^* J$$

that is $-\bar{\lambda}'$ is an eigenvalue of M. Similarly, if λ' is an eigenvalue of the pencil $M_1 - \lambda M_2$ with the left eigenvector y, then

$$y^* M_1 = \lambda' y^* M_2 \iff y^* M_1 J M_2^* = \lambda' y^* M_2 J M_2^* \iff M_1 (J M_2^* y) = -\bar{\lambda}' M_2 (J M_2^* y)$$

implying that $-\bar{\lambda}'$ is an eigenvalue as well. See [6] for a recent survey on Hamiltonian matrices and pencils. For the algorithms in this thesis it is essential that the eigenvalue solver used for Hamiltonian problems preserve the symmetry in the spectrum. An eigenvalue solver that is not designed to respect the eigenvalue

symmetry introduces real parts for imaginary eigenvalues in exact arithmetic because of rounding errors. Therefore tolerances would be needed to determine whether the real part computed is small enough so that the eigenvalue can be accepted as imaginary. Needless to say, determining the appropriate tolerance is a difficult task that depends on the conditioning of the real part of the eigenvalue as discussed in [45]. On the other hand, when an eigenvalue solver respecting the Hamiltonian symmetry of the eigenvalues is used, simple imaginary eigenvalues will remain on the imaginary axis under rounding errors avoiding the need for tolerances.

In [7] Hamiltonian eigenvalue solvers respecting the symmetry of the spectrum are given for real matrices and pencils. The key observation for the algorithms is that the eigenvalues of a Hamiltonian matrix M are minus and plus the square root of the eigenvalues of the matrix M^2 and the eigenvalues of a Hamiltonian pencil $M_1 - \lambda M_2$ are minus and plus the square root of the eigenvalues of the pencil $M_1 J^T M_1^T - \lambda M_2 J M_2^T$. The algorithm applies unitary transformations to M and $M_1 - \lambda M_2$ from left and right. Benner *et.al.* gave a procedure to construct a unitary matrix Q_1 and unitary symplectic matrices Q_2, Q_3, Q_4, Q_5 (a matrix Q is called unitary symplectic if $QJQ^* = I$) such that

$$Q_1^T M_2 Q_2 = \begin{bmatrix} \hat{M}_{11} & \hat{M}_{12} \\ 0 & \hat{M}_{22} \end{bmatrix}, Q_1^T M_1 Q_3 = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ 0 & \tilde{M}_{22} \end{bmatrix},$$

and

$$Q_4^T M Q_5 = \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix},$$

where $\hat{M}_{11}, \hat{M}_{22}^T, \tilde{M}_{11}, M_{11}$ are upper triangular matrices and $\tilde{M}_{22}^T, M_{22}^T$ are upper Hessenberg matrices. Notice that these transformations do not preserve the eigenvalues of M or $M_1 - \lambda M_2$; however, it can be easily derived that

$$Q_1^T M_2 J M_2^T Q_1 J = \begin{bmatrix} \hat{M}_{11} & \hat{M}_{12} \\ 0 & \hat{M}_{22} \end{bmatrix} \begin{bmatrix} -\hat{M}_{22}^T & \hat{M}_{12}^T \\ 0 & -\hat{M}_{11}^T \end{bmatrix},$$

$$Q_1^T M_1 J^T M_1^T Q_1 J = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ 0 & \tilde{M}_{22} \end{bmatrix} \begin{bmatrix} \tilde{M}_{22}^T & -\tilde{M}_{12}^T \\ 0 & \tilde{M}_{11}^T \end{bmatrix}$$

and

$$Q_4^T M^2 Q_4 = \begin{bmatrix} -M_{11} M_{22}^T & M_{11} M_{12}^T - M_{12} M_{11}^T \\ 0 & -M_{22} M_{11}^T \end{bmatrix}.$$

Therefore the eigenvalues of M^2 are same as the eigenvalues of $-M_{11} M_{22}^T$ with algebraic multiplicity doubled. On the other hand, the eigenvalues of the pencil $M_1 J^T M_1^T - \lambda M_2 J M_2^T$ are same as the generalized eigenvalues of

$\tilde{M}_{11}\tilde{M}_{22}^T + \lambda\hat{M}_{11}\hat{M}_{22}^T$. The eigenvalues of $-M_{11}M_{22}^T$ and $\tilde{M}_{11}\tilde{M}_{22}^T + \lambda\hat{M}_{11}\hat{M}_{22}^T$ can be computed via periodic QR and periodic QZ algorithms without forming the products. The eigenvalues of M are plus and minus square roots of the eigenvalues of $-M_{11}M_{22}^T$ and the eigenvalues of $M_1 - \lambda M_2$ are plus and minus square roots of the eigenvalues of $\tilde{M}_{11}\tilde{M}_{22}^T + \lambda\hat{M}_{11}\hat{M}_{22}^T$. The algorithm can be extended to complex matrices or pencils by replacing the matrix and the pencil with the real ones of double the size of the original problems.

A more desirable property of a Hamiltonian eigenvalue solver is to preserve not only the eigenvalue symmetry but also the Hamiltonian structure of the input matrix or pencil. Formally, a structure-preserving backward stable method for the matrix M would return the eigenvalues of

$$\tilde{M} = M + E \tag{A.2}$$

where $E, \tilde{M}$ are Hamiltonian and $\|E\| = O(\delta_{mach}\|M\|)$. A structure-preserving backward stable generalized Hamiltonian eigenvalue solver computing the eigenvalues of $M_1 - \lambda M_2$ returns the eigenvalues of the nearby Hamiltonian pencil $\tilde{M}_1 - \lambda\tilde{M}_2$

$$\tilde{M}_1 = M_1 + E_1 \tag{A.3}$$
$$\tilde{M}_2 = M_2 + E_2 \tag{A.4}$$

with $\|E_1\| = O(\delta_{mach}\|M_1\|)$ and $\|E_2\| = O(\delta_{mach}\|M_2\|)$. In this thesis our analysis to determine the backward error of the algorithms always assumes the usage of the eigenvalue solvers preserving the Hamiltonian structure.

Van Loan suggested a structure-preserving algorithm in [53]. However, this method suffers from the fact that half of the precision may be lost and is not backward stable in general. Recently Chu *et.al.* presented structure-preserving algorithms that are backward stable for nonimaginary eigenvalues of Hamiltonian matrices or pencils [18]. Since our algorithms are specifically based on the extraction of the imaginary eigenvalues, at the moment the most suitable Hamiltonian eigenvalue solvers for our purpose are the implementations of the algorithms in [7] included in the HAPACK library for structured eigenvalue problems [5].

A.2 Symplectic eigenvalue problems

A complex matrix M of size $2n$ is called *symplectic* if it satisfies the property

$$MJM^* = J,$$

while a pencil $M_1 - \lambda M_2$ is called *symplectic* if the equality

$$M_1 J M_1^* = M_2 J M_2^*$$

holds. Additionally we call a pencil $M_1 - \lambda M_2$ $*$-symplectic if the pencil $M_1^* - \lambda M_2^*$ is symplectic. The eigenvalues of symplectic matrices and pencils are symmetric with respect to the unit circle. For a symplectic matrix M with the eigenvalue λ and the left eigenvector y associated with λ,

$$y^*M = \lambda y^* \Longleftrightarrow y^*MJM^* = \lambda y^*JM^* \Longleftrightarrow y^*J = \lambda y^*JM^* \Longleftrightarrow 1/\bar{\lambda}(Jy) = M(Jy),$$

so $1/\bar{\lambda}$ is an eigenvalue as well (note that the symplectic property implies that M is nonsingular). For a symplectic pencil $M_1 - \lambda M_2$ with the eigenvalue λ and the associated left eigenvector y,

$$y^*M_1 = \lambda y^*M_2 \Longleftrightarrow y^*M_1JM_1^* = \lambda y^*M_2JM_1^* \Longleftrightarrow 1/\bar{\lambda}M_2(JM_2^*y) = M_1(JM_2^*y)$$

which means that $1/\bar{\lambda}$ is an eigenvalue as well. (A zero eigenvalue is paired with an infinite eigenvalue and vice versa.)

Some of our algorithms are based on the extraction of the unit eigenvalues of symplectic or $*$-symplectic pencils. To solve symplectic generalized eigenvalue problems we can reduce the pencil $M_1 - \lambda M_2$ to a generalized Hamitonian eigenvalue problem via a Cayley transformation, *i.e.* the pencil $M_1 - \lambda M_2$ is symplectic if and only if the pencil

$$M_{1H} - \lambda M_{2H} = (M_1 + M_2) - \lambda(M_1 - M_2) \tag{A.5}$$

is Hamiltonian. We can then benefit from a Hamiltonian eigenvalue solver applied to the pencil $M_{1H} - \lambda M_{2H}$ and transform back the eigenvalues. Clearly if λ is an eigenvalue of the pencil $M_{1H} - \lambda M_{2H}$, then $\frac{-(1+\lambda)}{1-\lambda}$ is an eigenvalue of the symplectic pencil $M_1 - \lambda M_2$. (Corresponding to each eigenvalue equal to one of the Hamiltonian pencil, the sympletic pencil has an eigenvalue at ∞ and vice versa.) The imaginary eigenvalues of the Hamiltonian pencil correspond to unit eigenvalues of the symplectic pencil. Thus using the algorithm in [7] described in the previous section and Cayley transformation, we can obtain eigenvalues of unit modulus reliably. In practice we use the implementation in HAPACK for Hamiltonian pencils followed by the transformation of the Hamiltonian eigenvalues into symplectic ones.

A.3 Even-odd polynomial eigenvalue problems

The matrix polynomial $P(\lambda)$ defined as

$$P(\lambda) = \sum_{j=0}^{k} \lambda^j K_j$$

with the complex adjoint

$$P^*(\lambda) = \sum_{j=0}^{k} \lambda^j K_j^*$$

is called *even*, *odd*, $*$-*even* or $*$-*odd* if it satisfies the property $P(\lambda) = P(-\lambda)$, $P(\lambda) = -P(-\lambda)$, $P^*(\lambda) = P(-\lambda)$ or $P^*(\lambda) = -P(-\lambda)$, respectively. The algorithms for higher-order dynamical systems in this thesis are based on the capability of extracting the imaginary eigenvalues of $*$-even matrix polynomials. Suppose that λ is an eigenvalue of the $*$-even or the $*$-odd matrix polynomial P and x is an associated right eigenvector; then

$$P(\lambda)x = x^*(P(\lambda))^* = x^*P^*(\bar{\lambda}) = x^*P(-\bar{\lambda}) = 0.$$

Therefore the eigenvalues of $P(\lambda)$ are either imaginary or in pairs $(\lambda, -\bar{\lambda})$. Similarly the eigenvalues of an even and a odd matrix polynomial are either imaginary or in pairs $(\lambda, -\lambda)$.

In general it is convenient to solve a polynomial eigenvalue problem via *linearization*, which is a procedure to replace the polynomial eigenvalue problem of size n with a generalized eigenvalue problem $\mathcal{H} + \lambda\mathcal{N}$ of size $kn \times kn$ with the same set of eigenvalues. Then the polynomial eigenvalues can be retrieved by a generalized eigenvalue solver applied to $\mathcal{H}+\lambda\mathcal{N}$. The most popular way of reducing a polynomial eigenvalue problem to a generalized eigenvalue problem is via the companion forms [46]

$$\mathcal{H}_{c1} + \lambda\mathcal{N}_{c1} = \begin{bmatrix} K_{k-1} & \dots & K_1 & K_0 \\ -I & & 0 & 0 \\ & \ddots & & \\ 0 & & -I & 0 \end{bmatrix} + \lambda \begin{bmatrix} K_k & 0 & & 0 \\ 0 & I & & \\ & & \ddots & \\ 0 & 0 & & I \end{bmatrix}$$

and

$$\mathcal{H}_{c2} + \lambda\mathcal{N}_{c2} = \begin{bmatrix} K_{k-1} & -I & & 0 \\ \vdots & & \ddots & \\ K_1 & & & -I \\ K_0 & & & 0 \end{bmatrix} + \lambda \begin{bmatrix} K_k & 0 & & 0 \\ 0 & I & & 0 \\ & & \ddots & \\ 0 & 0 & & I \end{bmatrix}$$

But in theory any transformation satisfying

$$F(\mathcal{H} + \lambda\mathcal{N})G = \begin{bmatrix} P(\lambda) & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix}$$

with nonsingular $F, G \in \mathbb{C}^{4nk \times 4nk}$ achieves the linearization task. For a detailed discussion on the solution of the polynomial eigenvalue problems, we refer to the book [46]. For the special case of quadratic eigenvalue problems the survey paper [67] is a comprehensive reference.

Here we focus on structured well-conditioned linearizations. In [56] vector spaces of linearizations that are generalizations of the companion forms are introduced. Let $\Lambda = [\lambda^{k-1} \ \ldots \ \lambda_1 \ \lambda_0]^T$. It is straightforward to verify that the companion forms satisfy the equalities

$$(\mathcal{H}_{c1} + \lambda \mathcal{N}_{c1})(\Lambda \otimes I) = e_1 \otimes P(\lambda)$$
$$(\Lambda \otimes I)(\mathcal{H}_{c2} + \lambda \mathcal{N}_{c2}) = e_1^T \otimes P(\lambda),$$

where $e_1 \in \mathbb{C}^k$ is the column vector of zeros except for the first entry, which is equal to one. By replacing the vectors e_1 appearing on the right-hand sides in the equations above with arbitrary vectors, the vector spaces

$$\mathbb{L}_1(P) = \{Y + \lambda X \ : \ \exists v \in \mathbb{C}^k \ (Y + \lambda X)(\Lambda \otimes I) = v \otimes P(\lambda)\}$$
$$\mathbb{L}_2(P) = \{Y + \lambda X \ : \ \exists w \in \mathbb{C}^k \ (Y + \lambda X)(\Lambda \otimes I) = w^T \otimes P(\lambda)\}$$

are obtained [56]. For a pencil $Y + \lambda X$ in $\mathbb{L}_1(P)$ such that the equality $(Y + \lambda X)(\Lambda \otimes I) = v \otimes P(\lambda)$ holds, the vector v is called a right ansatz vector, while for a pencil $Y + \lambda X$ in $\mathbb{L}_2(P)$ such that the equality $(\Lambda \otimes I)^T (Y + \lambda X) = w^T \otimes P(\lambda)$ holds, the vector w is called a left ansatz vector. In [56] it has been shown that any pencil in $\mathbb{L}_1(P)$ or $\mathbb{L}_2(P)$ is a linearization for a regular P if and only if the pencil is regular. This condition, though easy to state, is hard to interpret in terms of the input polynomials. The intersection of these two vector spaces

$$\mathbb{DL}(P) = \mathbb{L}_1(P) \cap \mathbb{L}_2(P)$$

has nicer properties. It turns out that any pencil in $\mathbb{DL}(P)$ has the left ansatz vector and the right ansatz vector equal to each other (see Theorem 5.3 in [56]). Indeed there is a unique pencil in $\mathbb{DL}(P)$ with a given ansatz vector. A pencil $Y + \lambda X \in \mathbb{DL}(P)$ with the ansatz vector v is a linearization for a regular P if and only if the set of roots of the scalar polynomial

$$p_v(x) = v_1 x^{k-1} + v_2 x^{k-2} + \cdots + v_k \tag{A.6}$$

and the set of eigenvalues of the matrix polynomial P are disjoint. Furthermore, the eigenvalues of the polynomial P and the eigenvalues of a linearization have different condition numbers and according to [35] the smaller the distance from a given root of $p_v(x)$ to the closest eigenvalue of $P(\lambda)$, the more ill-conditioned the corresponding eigenvalue of the linearization. Another desirable property

of a linearization $Y + \lambda X$ in $\mathbb{DL}(P)$ is that any of its left and right eigenvectors corresponding to finite eigenvalues are in the form $\bar{\Lambda} \otimes y$ and $\Lambda \otimes x$, respectively, where y and x are left and right eigenvectors of P. Therefore the eigenvectors of P can be constructed easily from those of the linearization $Y + \lambda X$.

As far as the algorithms in this thesis are concerned, the accurate computation of the exact imaginary eigenvalues of a $*$-even matrix polynomial without introducing real parts is the key property we seek. For this purpose the linearization must ideally have the $*$-even structure as well, that is we aim for a linearization $\mathcal{H} + \lambda\mathcal{N}$ where $\mathcal{H}$ is Hermitian and $\mathcal{N}$ is skew-Hermitian. In $\mathbb{DL}(P)$ there is no Hermitian/skew-Hermitian pencil. However, such pencils exist in $\mathbb{L}_1(P)$. Indeed in [55] it was shown that any pencil $Y + \lambda X \in \mathbb{L}_1(P)$ such that $(\Sigma \otimes I)(Y + \lambda X) \in \mathbb{DL}(P)$ with the ansatz vector v satisfying $\Sigma v = \bar{v}$ for

$$\Sigma = \begin{bmatrix} (-1)^{k-1}I & 0 & & \\ 0 & (-1)^{k-2}I & & \\ & & \ddots & \\ & & & I \end{bmatrix}$$

is a Hermitian/skew-Hermitian pencil.

Assuming that we approximately know the regions that contain the eigenvalues of P *a priori*, we can choose k roots away from these regions. The coefficients of the polynomial $p_v(x)$ with these roots provide us the ansatz vector v, which must also be forced to satisfy $\Sigma v = \bar{v}$. Then the unique linearization $Y + \lambda X \in \mathbb{DL}(P)$ with the ansatz vector v can be constructed and the pencil $(\Sigma \otimes I)(Y + \lambda X) = \mathcal{H} + \lambda\mathcal{N}$ with the right ansatz vector $\bar{v}$ is a Hermitian/skew-Hermitian linearization that approximately preserves the conditioning of the eigenvalues of P. Now the pencil $\mathcal{H} + \lambda\mathcal{N}i$ is a Hermitian/Hermitian pencil with the eigenvalue λ corresponding to each eigenvalue $i\lambda$ of P. The QZ algorithm with special care will typically return the eigenvalues of $\mathcal{H} + \lambda\mathcal{N}i$ in conjugate pairs $(\lambda, \bar{\lambda})$ which correspond to the pair of eigenvalues $(\lambda_P, -\bar{\lambda}_P)$ of the polynomial P, where $\lambda_P = i\lambda$. When P is real and the linearization $\mathcal{H} + \lambda\mathcal{N}$ is real, after deflating the infinite eigenvalues of $\mathcal{N}$, denote the resulting Hermitian/skew-Hermitian pencil by $\mathcal{H}_1 + \lambda\mathcal{N}_1$. Now the skew-symmetric matrix $\mathcal{N}_1$ can be factorized as $\mathcal{L}_1 J \mathcal{L}_1^T$ [4]. Therefore the Hermitian/skew-Hermitian generalized eigenvalue problem can be converted into the standard Hamiltonian eigenvalue problem $J\mathcal{L}_1^{-1}\mathcal{H}_1\mathcal{L}_1^{-T} - \lambda I$ which can be solved using algorithm [7] in HAPACK.

A.4 Palindromic polynomial eigenvalue problems

Let us define the reverse and $*$-reverse of a matrix polynomial as $\mathrm{rev}(P(\lambda)) = \sum_{j=0}^{k} \lambda^j K_{k-j}$ and $\mathrm{rev}(P^*(\lambda)) = \sum_{j=0}^{k} \lambda^j K_{k-j}^*$. We call a matrix polynomial P *palindromic* or $*$-*palindromic* if the properties $P(\lambda) = \mathrm{rev}(P(\lambda))$ or $P(\lambda) = \mathrm{rev}(P^*(\lambda))$ hold, respectively. Some of the algorithms for higher-order systems depend on the extraction of the unit eigenvalues of $*$-palindromic matrix polynomials. A $*$-palindromic matrix polynomial has eigenvalue symmetry with respect to the unit circle as for each finite non-zero eigenvalue λ with the right eigenvector x,

$$P(\lambda)x = 0 \Longleftrightarrow \mathrm{rev}P(1/\lambda)x = 0 \Longleftrightarrow x^*\mathrm{rev}P^*(1/\bar{\lambda}) = 0 \Longleftrightarrow x^*P(1/\bar{\lambda}) = 0,$$

so the scalar $1/\bar{\lambda}$ is an eigenvalue as well. (An infinite eigenvalue pairs with zero.) Similarly, the eigenvalues of a palindromic matrix polynomial are in pairs $(\lambda, 1/\lambda)$.

To solve a $*$-palindromic polynomial eigenvalue problem by preserving the symmetry of the spectrum, the palindromic matrix polynomial P can be transformed into the $*$-even matrix polynomial P_e via the Cayley transformation

$$P_e(\lambda) = (\lambda + 1)^k P\left(\frac{1-\lambda}{1+\lambda}\right). \tag{A.7}$$

Notice that the Cayley transformation above maps the eigenvalues -1 and ∞ to each other. Any finite eigenvalue $\lambda \neq -1$ of P_e corresponds to the eigenvalue $\frac{1-\lambda}{1+\lambda}$ of P. In particular, an imaginary eigenvalue of P_e matches with a unit eigenvalue of P and vice versa.

To summarize, to retrieve the unit eigenvalues of a $*$-palindromic polynomial, we can simply perform the Cayley transformation (A.7). Then we compute the eigenvalues of the $*$-even polynomial P_e as discussed in the previous section and transform back the $*$-even polynomial eigenvalues into the $*$-palindromic eigenvalues.

Basic Notation

$\mathbb{R}$	:	real numbers
$\mathbb{C}$	:	complex numbers
$\mathbb{Z}$	:	integers
$\mathbb{R}_+$	:	nonnegative real numbers
$\mathbb{C}_+$	:	complex numbers with nonnegative real parts
$\mathbb{C}^n$	:	vectors of complex numbers of size n
$\mathbb{R}^n$	:	vectors of real numbers of size n
$\mathbb{R}^n_+$	:	vectors of nonnegative real numbers of size n
$\mathbb{C}^{n\times m}$	:	complex $n \times m$ matrices
$\mathbb{R}^{n\times m}$	:	real $n \times m$ matrices
$H(A)$	:	Hermitian part $\frac{A+A^*}{2}$ of the matrix A
$N(A)$	:	skew-Hermitian part $\frac{A-A^*}{2}$ of the matrix A
$\det(A)$	:	determinant of the matrix A
$\|A\|$	:	2-norm of the matrix A
$\|A\|_F$	:	Frobenius norm of the matrix A
$\Lambda(A)$	:	spectrum of the matrix A
$\Lambda(P)$	:	spectrum of the matrix polynomial P
$\Lambda_\epsilon(A)$	:	ϵ-pseudospectrum of the matrix A
$\Lambda_\epsilon(P,\gamma)$	:	ϵ-pseudospectrum of the matrix polynomial P with the scaling vector γ
$F(A)$	:	field of values of the matrix A
$F(P)$	:	field of values of the matrix polynomial P
$\sigma(A)$	:	set of singular values of the matrix A
$\lambda_{\max}(A)$	:	largest eigenvalue of the Hermitian matrix A
$\lambda_{\min}(A)$	:	smallest eigenvalue of the Hermitian matrix A
$\lambda_j(A)$	:	jth smallest eigenvalue of the Hermitian matrix A
$\sigma_{\min}(A)$	:	$\min(n,m)$th largest singular value of the $n\times m$ matrix A
$\alpha(A)$	:	spectral abscissa of the matrix A
$\alpha(P)$	:	spectral abscissa of the matrix polynomial P
$\rho(A)$	:	spectral radius of the matrix A
$\rho(P)$	:	spectral radius of the matrix polynomial P

$\alpha_F(A)$	:	numerical abscissa of the matrix A
$r(A)$	:	numerical radius of the matrix A
$\alpha_\epsilon(A)$	:	ϵ-pseudospectral abscissa of the matrix A
$\alpha_\epsilon(P,\gamma)$	:	ϵ-pseudospectral abscissa of the matrix polynomial P with the scaling vector γ
$\rho_\epsilon(A)$	:	ϵ-pseudospectral radius of the matrix A
$\rho_\epsilon(P,\gamma)$	:	ϵ-pseudospectral radius of the matrix polynomial P with the scaling vector γ
$\beta_c(A)$	:	continuous distance to instability of the matrix A
$\beta_c(P,\gamma)$	:	continuous distance to instability of the matrix polynomial P with the scaling vector γ
$\beta_d(A)$	:	discrete distance to instability of the matrix A
$\beta_d(P,\gamma)$	:	discrete distance to instability of the matrix polynomial P with the scaling vector γ
$\tau(A,B)$	:	distance to uncontrollability of the matrix pair (A,B)
$\tau(P,B,\gamma)$	:	distance to uncontrollability of the matrix polynomial P and matrix B with the scaling vector γ
$\mathrm{Re}\ z$	:	real part of the complex number z
$\mathrm{Im}\ z$	:	imaginary part of the complex number z
$\lvert z\rvert$	:	modulus of the complex number z
$A \otimes B$	:	Kronecker product of the matrices A and B
$\mathbf{vec}(A)$	:	column vector obtained by stacking up the columns of the matrix A
$\mathbf{rvec}(A)$	:	row vector obtained by concatenating the rows of the matrix A
δ_{mach}	:	machine precision which is equal to 2^{-53} in IEEE double precision floating point arithmetic
$\mathbb{C}_g$	:	open set representing the stable region
$\mathbb{C}_b$	:	closed set representing the unstable region
∂C_b	:	boundary of the unstable region

Bibliography

[1] E. Anderson, S. Ostrouchov, D. Sorensen, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, and A. McKenney. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992. 88, 100

[2] O. Axelsson, H. Lu, and B. Polman. On the numerical radius of matrices and its application to iterative solution methods. *Linear and Multilinear Algebra*, 37:225–238, 1994. 62

[3] R.H. Bartels and G.W. Stewart. Solution of the equation AX+XB=C. *Comm. ACM*, 15:820–826, 1972. 100

[4] P. Benner, R. Byers, H. Fassbender, V. Mehrmann, and D. Watkins. Cholesky-like factorizations of skew-symmetric matrices. *Electr. Trans. Num. Anal.*, 11:85–93, 2000. 141

[5] P. Benner and D. Kressner. HAPACK – software for structured eigenvalue problems. Available from `http://www.tu-chemnitz.de/mathematik/hapack/`. 137

[6] P. Benner, D. Kressner, and V. Mehrmann. Skew-Hamiltonian and Hamiltonian eigenvalue problems: Theory, algorithms and applications. In *Proccedings of ApplMath03, Brijuni (Croatia)*, pages 66–75. Kluwer, June 23-27, 2003. 135

[7] P. Benner, V. Mehrmann, and H. Xu. A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils. *Numerische Mathematik*, 78:329–358, 1998. 64, 136, 137, 138, 141

[8] J.F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000. 44

[9] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing

its L_∞-norm. *Systems and Control Letters*, 15:1–7, 1990. 13, 15, 16, 28, 30, 39, 47, 63, 64, 78

[10] N.A. Bruinsma and M. Steinbuch. A fast algorithm to compute the H_∞-norm of a transfer function matrix. *Systems and Control Letters*, 14:287–293, 1990. 28

[11] J.V. Burke, A.S. Lewis, and M.L. Overton. Optimization and pseudospectra, with applications to robust stability. *SIAM Journal on Matrix Analysis*, 25(1):80–104, 2003. 22, 41, 43, 44, 45

[12] J.V. Burke, A.S. Lewis, and M.L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA Journal of Numerical Analysis*, 23(3):359–375, 2003. 13, 14, 15, 16, 19, 24, 30, 41, 47, 51, 52, 53

[13] J.V. Burke, A.S. Lewis, and M.L. Overton. Pseudospectral components and the distance to uncontrollability. *SIAM J. Matrix Analysis Appl.*, 26:350–361, 2004. 13, 14, 75, 76, 78, 81, 82, 86, 100, 106, 113, 125

[14] J.V. Burke, A.S. Lewis, and M.L. Overton. Software for the distance to uncontrollability using Gu's scheme, 2005. `http://www.cs.nyu.edu/faculty/overton/software/uncontrol/`. 86

[15] R. Byers. A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM Journal on Scientific and Statistical Computing*, 9:875–881, 1988. 16, 28, 45, 46, 56

[16] R. Byers. Detecting nearly uncontrollable pairs. In *Numerical Methods Proceedings of the International Symposium MTNS-89*, volume III, pages 447–457. M.A. Kaashoek, J.H. van Schuppen, and A.C.M. Ran, eds., Springer-Verlag, 1990. 14, 78, 79, 80, 125

[17] R. Byers. The descriptor controllability radius. In *Numerical Methods Proceedings of the International SymposiumMTNS-93*, volume II, pages 85–88. Uwe Helmke, Reinhard Mennicken, and Hosef Saurer, eds., Akademie Verlag, Berlin, 1993. 113

[18] D. Chu, X. Liu, and V. Mehrmann. A numerically strongly stable method for computing the Hamiltonian Schur form. September 2004. Technical report 24/2004, Institut für Mathematik, Technische Universität Berlin. 137

[19] K-W. Chu. Controllability of descriptor systems. *Internat. J. Control*, 46:1761–1770, 1987. 113

[20] K-W. Chu. A controllability condensed form and a state feedback pole assignment algorithm or descriptor systems. *IEEE Trans. Automat. Control*, 33:366–369, 1988. 113

[21] J.W. Demmel. A counterexample for two conjectures about stability. *IEEE Trans. Auto. Cont.*, 32:340–342, 1987. 32, 44

[22] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: a Convex Approach*. Springer-Verlag, New York, 2000. 3

[23] M. Eiermann. Field of values and iterative methods. *Linear Algebra and Its Applications*, 180:167–197, 1993. 62

[24] R. Eising. The distance between a system and the set of uncontrollable systems. In *Memo COSOR 82-19, Eindhoven Univ. Technol., Eindhoven, The Netherlands*, 1982. 75

[25] R. Eising. Between controllable and uncontrollable. *System Control Lett.*, 4:263–264, 1984. 75

[26] M. Gao and M. Neumann. A global minimum search algorithm for estimating the distance to uncontrollability. *Linear Algebra Appl.*, 188-189:305–350, 1993. 79

[27] L. Gaul and N. Wagner. Eigenpath dynamics of non-conservative mechanical systems such as disc brakes. In *IMAC XXII Dearborn, Michigan*, Jaunary 26-29 2004. 126

[28] Y. Genin, R. Stefan, and P. Van Dooren. Real and complex stability radii of polynomial matrices. *Linear Algebra Appl.*, 351-352:381–410, 2002. 8, 10

[29] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982. 12

[30] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore MD, 1996. 82

[31] M. Gu. New methods for estimating the distance to uncontrollability. *SIAM J. Matrix Analysis Appl.*, 21(3):989–1003, 2000. 13, 14, 75, 76, 78, 81, 82, 84, 86, 101, 106, 113, 125

[32] M. Gu, E. Mengi, M.L. Overton, J. Xia, and J. Zhu. Fast methods for estimating the distance to uncontrollability. *SIAM J. Matrix Analysis Appl.*, 2006. to appear. 78, 88

[33] C. He. Estimating the distance to uncontrollability: A fast method and a slow one. *Systems Control Lett.*, 26:275–281, 1995. 79

[34] C. He and G.A. Watson. An algorithm for computing the numerical radius. *IMA Journal of Numerical Analysis*, 17(3):329–342, July 1997. 13, 39, 62, 63

[35] N.J. Higham, D.S. Mackey, and F. Tisseur. The conditioning of linearizations of matrix polynomials. *SIAM J. Matrix Analysis Appl.*, 2006. to appear. 140

[36] N.J. Higham and F. Tisseur. More on pseudospectra for polynomial eigenvalue problems and applications in control theory. *Linear Algebra and its Applications*, 351-352:435–453, 2002. 2

[37] D. Hinrichsen and B. Kelb. Spectral value sets: a graphical tool for robustness analysis. *Systems Control Lett.*, 21:127–136, 1993. 8

[38] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. 23, 56, 78

[39] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. 12, 61, 62, 85

[40] G. Hu and E.J. Davison. Real controllability/stabilizability radius of LTI systems. *IEEE Trans. Automatic Cont.*, 49(2):254–257, 2004. 12

[41] I. Jonsson and B. Kågström. Recursive blocked algorithm for solving triangular systems-part I: one-sided and coupled Sylvester-type matrix equations. *ACM Trans. Math. Software*, 22(4):392–415, 2002. 100

[42] R.E. Kalman. Mathematical description of linear systems. *SIAM J. Control Optim.*, 1:152–192, 1963. 2

[43] M. Karow. *Geometry of Spectral Value Sets*. Ph.D. Thesis, Mathematik and Informatik, Universitat Berlin, 2003. 8

[44] H.-O. Kreiss. Uber die stabilitatsdefinition fur differenzengleichungen die partielle differentialgleichungen approximieren. *BIT*, 2:153–181, 1962. 16, 40

[45] D. Kressner and E. Mengi. Structure-preserving eigenvalue solvers for robust stability and controllability estimates, 2006. Submitted to IEEE conference on control and decision, 2006. 136

[46] P. Lancaster. *Lambda-Matrices and Vibrating Systems*. Pergamon Press, Oxford, UK, 1966. 139, 140

[47] P. Lancaster. *Theory of Matrices*. Academic Press, New York, 1969. 104

[48] R.B. Lehoucq, K. Maschoff, D. Sorensen, and C. Yang. ARPACK software package, 1996. `http://www.caam.rice.edu/software/ARPACK/`. 88

[49] R.B. Lehoucq, D. Sorensen, and C. Yang. *ARPACK Users' Guide*. SIAM, Philadelphia, 1998. 88

[50] A.S. Lewis. Private communication, 2004. 22, 42

[51] A.S. Lewis. Robust regularization. Technical report, Simon Fraser University, 2002. Submitted to Mathematical Programming. 22, 42

[52] C. Li and L. Rodman. Numerical range of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 15(4):1256–1265, 1994. 12

[53] C.F. Van Loan. A symplectic method for approximating all eigenvalues of a Hamiltonian matrix. *Linear Algebra Applications*, 61:233–251, 1984. 137

[54] C.F. Van Loan. How near is a stable matrix to an unstable matrix? *Contemporary Math.*, 47:465–477, 1985. 11

[55] D.S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Palindromic polynomial eigenvalue problems: Good vibrations from good linearizations. *Technical Report, DFG Research Center MATHEON, Mathematics for Key Technologies*, 2006. submitted to SIAM J. Matrix Analysis Appl. 141

[56] D.S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *Technical Report, DFG Research Center MATHEON, Mathematics for Key Technologies*, 2006. submitted to SIAM J. Matrix Analysis Appl. 140

[57] E. Mengi. On the estimation of the distance to uncontrollability for higher order systems. 2006. submitted to SIAM J. Matrix Analysis Appl. 113

[58] E. Mengi. Software for robust stability and controllability measures, 2006. `http://www.cs.nyu.edu/~mengi/robuststability.html`. 3, 14, 106, 129

[59] E. Mengi and M.L. Overton. Algorithms for the computation of the pseudospectral radius and the numerical radius of a matrix. *IMA J. of Numer. Anal.*, 25(4):648–669, 2005. 39

[60] J. Moro, J.V. Burke, and M.L. Overton. On the Lidskii-Vishik-Lyusternik perturbation theory for eigenvalues of matrices with arbitrary Jordan block. *SIAM J. Matrix Anal. Appl.*, 18(4):793–817, 1997. 104

[61] C.C. Paige. Properties of numerical algorithms relating to controllability. *IEEE Trans. Automat. Control*, AC-26:130–138, 1981. 13

[62] G. Pappas and D. Hinrichsen. Robust stability of linear systems described by higher order dynamic equations. *IEEE Trans. Automat. Control*, 38:1430–1435, 1993. 8

[63] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice Hall, Englewood Cliffs, NJ, 1980. 102

[64] C. Pearcy. An elementary proof of the power inequality for the numerical radius. *Michigan Math Journal*, 13:284–291, 1966. 61

[65] L. Qiu, B. Bernhardsson, A. Rantzer, E.J. Davison, P.M. Young, and J.C. Doyle. A formula for computation of the real stability radius. *Automatica*, 31:879–890, 1995. 8, 10

[66] F. Tisseur and N.J. Higham. Structured pseudospectra for polynomial eigenvalue problems with applications. *SIAM J. Matrix Anal. Appl.*, 23(1):187–208, 2001. 8

[67] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001. 11, 34, 140

[68] L.N. Trefethen. Pseudospectra of matrices. In *Proceedings of the 14th Dundee Conference*, volume II, pages 234–266. D.F. Griffiths and G.A. Watson, edsa, Pitman Res. Notes Math. Ser. 260, Longman Scientific and Technical, Harlow, UK, 1992. 11

[69] L.N. Trefethen. Computation of pseudospectra. *Acta Numerica*, pages 247–295, 1999. 11, 134

[70] L.N. Trefethen and M. Embree. *Spectra and Pseudospectra*. Princeton University Press, 2005. 8, 11, 40

[71] L.N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, 1997. 22

[72] E. Wegert and L.N. Trefethen. From the Buffon needle problem to the Kreiss matrix. *Amer. Math Monthly*, 101:132–139, 1994. 40

[73] T.G. Wright. Eigtool: a graphical tool for nonsymmetric eigenproblems. Oxford University Computing Laboratory. `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`. 11, 16, 19, 67, 106, 134

[74] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, 1996. 8