The spectrum of delay-differential equations: numerical methods, stability and perturbation

Von der Carl-Friedrich-Gauß-Fakultät
Technische Universität Carolo-Wilhelmina zu
Braunschweig

zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von Elias Jarlebring
geboren am 3. Februar 1978
in Vännäs, Schweden

Eingereicht am:	01.02.2008
Mündliche Prüfung am:	28.05.2008
Referentin:	Prof. Dr. Heike Faßbender, TU Braunschweig
Korreferent:	Prof. Dr. Achim Ilchmann, TU Ilmenau
Korreferent:	Prof. Dr. Daniel Kreßner, ETH Zürich

2008

Notation

$\mathbb{R}$	real numbers
$\mathbb{R}_+$	positive real numbers
$\mathbb{R}_-$	negative real numbers
$\mathbb{Z}$	integer numbers
$\mathbb{Q}$	rational numbers
$\mathbb{C}$	complex numbers
$\mathbb{C}_+$	open right half plane
$\mathbb{C}_-$	open left half plane
$\mathbb{D}, \partial\mathbb{D}$	open unit disc ($\mathbb{D} := \{z \in \mathbb{C} : \lvert z\rvert < 1\}$), unit circle
∂C	boundary of the set C
$\operatorname{clos} C$	closure of the set C, $\operatorname{clos} C = \partial C \cup C$
i	imaginary unit, $i^2 = -1$
$\operatorname{Re} z$	real part of the complex number z
$\operatorname{Im} z$	imaginary part of the complex number z
$l(z_1, z_2)$	straight line connecting $z_1, z_2 \in \mathbb{C}$ not including endpoints
$\operatorname{Arg} z$	principal branch argument of the complex number z
$\bar{z}$	complex conjugate of the complex number z
A^*	complex conjugate transpose of the matrix A
$\sigma(A), \sigma(\Sigma)$	spectrum of the matrix A or system Σ
$\sigma(G)$	solutions of $s \in \sigma(G(s))$ where $G : \mathbb{C} \to \mathbb{C}^{n\times n}$
$\sigma(A, B)$	solutions of the generalized eigenvalue problem $\det(A - \lambda B) = 0$
$\sigma_{\min}(A)$	the smallest singular value of the matrix A
$\lVert \cdot \rVert$	Eucledian or spectral norm
$\kappa(A)$	condition number of matrix A
$r_\sigma(A)$	spectral radius of matrix A
$\operatorname{atan}\binom{a}{b}$	four quadrant inverse of tangent, $\operatorname{atan}\binom{a}{b} = \operatorname{Arg}(b + ai)$
$\operatorname{sgn}(z)$	sign of complex number z, $\operatorname{sgn}(z) = z/\lvert z\rvert$ if $z \neq 0$
$\mathcal{O}(g(x))$	big O, $f(x) = \mathcal{O}(g(x)) \Leftrightarrow \exists M > 0, x_0 > 0 : \lvert f(x)\rvert < M\lvert g(x)\rvert \, \forall\, x > x_0$
ϵ_{mach}	machine precision, where not explicitly stated: $\epsilon_{\text{mach}} \approx 2.2 \cdot 10^{-16}$
$\otimes, \oplus$	Kronecker product, Kronecker sum
W_k	kth branch of the Lambert W function
$d_m(S_1, S_2)$	maxmin distance between sets S_1 and S_2
$d_H(S_1, S_2)$	Hausdorff distance between sets S_1 and S_2
$\mathcal{C}([a, b])$	A continuous function on the segment $t \in [-a, b]$

Acronyms

DDE	delay-differential equation
DEP	delay eigenvalue problem
IGD	infinitesimal generator discretization
LHP	left half-plane
LMI	linear matrix inequality
LMS	linear multistep
MS	Milne-Simpson
NP-hard	nondeterministic polynomial-time hard
PDDE	partial delay-differential equation
PDE	partial differential equation
PS	pseudospectral
RHP	right half-plane
RI	Rayleigh iteration
RII	residual inverse iteration
SOD	solution operator discretization
SRII	subspace accelerated residual inverse iteration
TDS	time-delay system

Contents

Abstract

Three types of problems related to *time-delay systems* are treated in this thesis. In our context, a time-delay system, or sometimes *delay-differential equation* (DDE), is a generalization of an ordinary differential equation (ODE) with constant coefficients. For DDEs, unlike ODEs, the derivative of the state at some time-point is not only dependent on the state at that time-point, but also on one or more previous states.

We first consider the problem of numerically computing the *eigenvalues* of a DDE, i.e., finding solutions of the *characteristic equation*, here referred to as the *delay eigenvalue problem*. Unlike standard ODEs, the characteristic equation of a DDE contains an exponential term. Because of this nonlinear term, the delay eigenvalue problem belongs to a class of problems referred to as *nonlinear eigenvalue problems*. An important contribution of the first part of this thesis is the application of a projection method for nonlinear eigenvalue problems, to the author's knowledge, previously not applied to the delay eigenvalue problem. We compare this projection method with other methods, suggested in the literature, and used in software packages. This includes methods based on discretizations of the *solution operator* and discretizations of the equivalent *boundary value problem formulation* of the DDE, i.e., the *infinitesimal generator*. We review discretizations based on, but not limited to, *linear multi-step*, *Runge-Kutta* and *spectral collocation*. The projection method is computationally superior to all of the other tested method for the presented large-scale examples. We give interpretations of the methods based on discretizations in terms of rational approximations of the exponential function or the logarithm. Some notes regarding a special case where the spectrum can be explicitly expressed are presented. The spectrum can be expressed with a formula containing a matrix version of the Lambert W func-

tion. The formula is not new. We clarify its range of applicability, and, by counter-example, show that it does not hold in general.

The second part of this thesis is related to exact stability conditions of the DDE. We find exact conditions on the delays such that the DDE has a purely imaginary eigenvalue. All those combinations of the delays such that there is a purely imaginary eigenvalue (called *critical delays*) are parameterized. That is, we construct a *mapping* consisting of computable expressions, from a simple mathematical object onto the set of all subsets of the critical delays. The mapping can be expressed explicitly with trigonometric functions for scalar DDEs. We find some new formulas and verify some formulas in the literature. In general, an evaluation of the map consists of solving a *quadratic eigenvalue problem* of squared dimension for non-scalar problems. The constructed eigenvalue problem is large, even for DDEs of moderate size. For that reason, we show how the computational cost for one evaluation of the map can be reduced. In particular we show that the matrix-vector product corresponding to the *companion linearization* of the quadratic eigenvalue problem can be computed by solving a *Lyapunov equation*. Most of the results in the chapter on critical delays are derived for retarded DDEs as well as neutral DDEs with an arbitrary number of delays.

The third and last part of this thesis is about perturbation results for nonlinear eigenvalue problems. We discuss some results in eigenvalue perturbation theory which can be generalized to nonlinear eigenvalue problems. A sensitivity formula for the movement of the eigenvalues extends nicely to nonlinear eigenvalue problems. We introduce a fixed point form for the nonlinear eigenvalue problem, and show that some methods in the literature can be interpreted as set-valued fixed point iterations. The convergence order of these types of iterations can be determined from an expression containing the left and right eigenvectors. We also use some results from *fixed point theory*. The famous result in perturbation theory referred to as *the Bauer-Fike theorem*, can be generalized to the nonlinear eigenvalue problem if we assume that the set-valued fixed point problem has a certain contraction property.

Zusammenfassung

In dieser Arbeit werden drei verschiedene Problemklassen im Bezug zu time-delay Systemen behandelt. Unter einem time-delay System, oder manchmal delay-differential equation (DDE), verstehen wir die Verallgemeinerung einer gewöhnlichen Differentialgleichung (ODE) mit konstanten Koeffizienten. Wobei im Gegensatz zu ODEs, bei DDEs die Ableitung zu jedem Zeitpunkt nicht nur vom aktuellen, sondern auch von einem oder mehreren vorhergehenden Zuständen abhängig ist.

Als erstes gehen wir auf die numerischen Berechnung der Eigenwerte von DDEs ein. Das heißt es sind Lösungen der charakterischen Gleichung zu finden. Dieses Problem wird im Folgenden als Delay-Eigenwertproblem (DEP) bezeichnet. Im Gegensatz zu ODEs enthält die charakteristische Gleichung einer DDE einen exponentiellen Term. Aufgrund des nichtlinearen Terms gehört das Delay-Eigenwertproblem zur Klasse der nichtlinearen Eigenwertprobleme. Ein wichtiger Beitrag dieser Arbeit ist die Anwendung einer Projektionsmethode für nichtlineare Eigenwertprobleme, welche bisher noch nicht auf DEPs angewendet wurde. Wir vergleichen diese Projektionsmethode mit anderen in der Literatur vorgeschlagenen und in Softwarepaketen verwendeten Verfahren. Dieser Vergleich schliesst Methoden der Diskretisierung des Lösungsoperators sowie der Diskretisierung des äquivalenten Randwertproblems ein. Dabei betrachten wir auf Diskretisierung basierende Methoden, wie lineare Mehrschrittverfahren, Runge-Kutta Verfahren und spectral collocation, näher. Es stellt sich heraus, dass die hier vorgestellte Projektionsmethode bedeutend bessere numerische Eigenschaften für die hier verwendeten großen Beispiele, als sämtliche andere getestete Verfahren besitzt. Zusätzlich treffen wir Aussagen über Diskretisierungsmethoden zur rationalen Approximation der Exponentialfunktion bzw. des Logarithmus. Des weiteren betrachten wir einen Spezialfall, bei welchem das Spektrum explizit mit Hilfe einer

Matrix-Version der Lambert W-Funktion dargestellt werden kann. Für diese an sich nicht neue Formel bestimmen wir einen möglichen Anwendungsbereich und zeigen durch ein Gegenbeispiel, dass diese nicht allgemein gilt.

Im zweiten Teil der Arbeit werden exakte Stabilitätsbedingungen von DDEs betrachtet. Für die Delays werden exakte Bedingungen so bestimmt, dass die DDE einen rein imaginären Eigenwert besitzt. Die Menge dieser Delays, sogenannte *kritische Delays*, wird parameterisiert. Das heißt, wir bilden eine aus berechenbaren Ausdrücken bestehende Abbildung von einem einfachen mathematischen Objekt auf die Menge aller Teilmengen kritischer Delays. Für skalare Probleme wird diese Abbildung mit trigonometrischen Funktionen explizit ausgedrückt. Dabei werden sowohl neue Formeln hergeleitet als auch bereits in der Literatur bekannte Formeln bestätigt. Für nicht skalare Probleme ist zur Auswertung der Abbildung das Lösen eines *quadratischen Eigenwertproblems* nötig, dessen Größe dem Quadrat der Dimension der DDE entspricht. Damit wird auch für DDEs kleiner und mittlerer Dimension das konstruierte Eigenwertproblem groß beziehungsweise sehr groß. Weiterhin werden möglichkeiten zur Reduktion des Rechenaufwandes der Auswertung der Abbildung diskutiert. Es wird gezeigt, dass das zur companion Linearisierung des quadratischen Eigenwertproblems gehörende Matrix-Vektor Produkt durch das Lösen einer Lyapunov-Gleichung berechnet werden kann. Die meisten Ergebnisse des Kapitels über kritische Delays sind speziell auf DDEs mit beliebiger Anzahl von Delays, sowohl retardierter als auch neutraler Form, anwendbar.

Der dritte und letzte Teil dieser Arbeit befasst sich mit der Störungstheorie von nichtlinearen Eigenwertproblemen. Hierin werden einige Aussagen über Ergebnisse der Eigenwertstörungstheorie, welche sich auf nichtlineare Eigenwertprobleme verallgemeinern lassen, getroffen. Unter anderem lässt sich eine Formel für die Sensitivität auf nichtlineare Eigenwertprobleme erweitern. Desweiteren wird eine Fixpunktform für nichtlineare Eigenwertprobleme vorgestellt, und gezeigt dass einige Methoden aus der Literatur als mengenwertige Fixpunktiterationen dargestellt werden können. Die Konvergenzordnung solcher Iterationen kann durch einen aus linkem und rechtem Eigenvektor bestehenden Ausdruck bestimmt werden. Ausserdem werden einige Ergebnisse aus der Fixpunkttheorie verwendet. Damit kann das in der Störungstheorie bekannte Bauer-Fike Theorem auf nichtlineare Eigenwertprobleme verallgemeinert werden, wenn angenommen wird, dass das mengenwertige Fixpunktproblem äquivalent zu einer Kontraktion ist.

Acknowledgements

This thesis would not have been possible without the assistance and support of friends and colleagues at the *technical university of Braunschweig* and elsewhere.

First of all, I have the pleasure to thank my supervisor, Heike Fassbender, for guidance and very valuable support throughout my years as a PhD student in Braunschweig. I am grateful for her outstanding efforts which have made this thesis possible.

Among the friends and colleagues, special thanks go to: Tobias Damm for very often providing constructive accurate criticism, my master thesis examiner and supervisor Axel Ruhe, KTH Stockholm, for discussions and allowing a nice time as a visitor at the numerical analysis group at KTH, Michiel E. Hochstenbach, for his enthusiasm, inspiration, discussions and advices, and my former master thesis supervisor Heinrich Voss, for discussions and valuable comments.

I am also grateful to many other colleagues and friends for technical discussions, and fruitful comments on programming issues, as well as drafts and final versions of this thesis: Matthias Bollhöfer, Dimitri Breda, Henrik Bäärnhielm, Johan Karlsson, Daniel Kressner, Volker Mehrmann, Wim Michiels, Timo Reis, Bor Plestenjak, Peter Stange and Fabian Wirth.

Chapter 1

Introduction

Many physical phenomena are described with mathematical models expressed as differential equations. That is, the derivative of the state at some time point is a function of the current state and the current time. The important special case that the derivative of the state is a linear combination of the state, i.e.,

$$\dot{x}(t) = Ax(t),\ A \in \mathbb{R}^{n\times n}$$

is treated in analysis text-books for undergraduate studies in the science subjects.

The topic of this thesis is a generalization of this ordinary differential equation (ODE). We allow an additional term in the state equation depending on the state at some previous time-point. Mostly, we consider linear delay-differential equations (DDEs) with one delay,

$$\dot{x}(t) = A_0x(t) + A_1x(t-h),$$

or multiple delays

$$\dot{x}(t) = A_0x(t) + \sum_{k=1}^{m} A_kx(t-h_k).$$

That is, the derivative of the state is a linear combination of the current state and one or more previous states.

In this thesis we discuss numerical tools and some theoretical results used to analyze DDEs.

Some applications

This generalization of the ODE is important, as it allows the mathematical treatment of models with delays. Indeed, many physical events do not occur instantaneously and can be modeled with delays. We mention some models with delays from engineering, physics and chemistry.

The electronic signal of the control of a robot takes some time to go from the controller to the robot arm. Similarly, if the controllers of the wing-rudders of an airplane are located in the cockpit, the controllers can only control the rudders with a certain delay. When a human driver on a high-way observes that the next car is breaking, he will hit the breaks after a certain reaction time. In the modelling of a high-way congestion, this reaction time influences the length of the congestion. Chemical reactions do normally not occur instantaneously. Suppose a manufacturer wants to produce a material with a customer specified material property. An (unpolluted) construction of a material with the given properties is sometimes only possible with an accurate control of the chemical process.

Delays are also relevant in more critical applications. The accurate modelling and control of nuclear reactors is crucial. The temperature of the inner part of nuclear reactor may not be available for measurement. If the temperature in the inner part rises, after some time (delay), the temperature of the surface of the reactor will also rise. Hence, only old information about the state is available for measurement and can be used to control the process.

Since delay-differential equations appear in a large number of fields in science. It is not surprising that it has received different names in different fields. For instance, the following terms are used for slight variations of DDEs, *time-delay systems*, *difference-differential equations*, *retarded systems*, *functional differential equations*[1].

In particular, there are many results on DDEs in the field of control and systems theory, functional analysis, mathematical biology, numerical analysis and analysis in general.

In this thesis we discuss results from several fields, but attempt a consistent presentation by using mostly the terminology and concepts of numerical linear algebra. In particular, we use the terminology of numerical methods for eigenvalue problems.

[1] DDEs is only one of many applications of functional differential equations.

An illustrative example

The main concepts of this thesis are illustrated with a simple example.

Even though not as critical as the delay effects in the control of an airplane or a nuclear reactor, the DDE can be used to describe the human being standing under the shower trying to reach his optimal shower-temperature by turning the shower tap (controller). We will illustrate some of the topics of this thesis with this simple, popular, motivating example known as the *hot shower problem*.

Example 1.1 (Hot shower problem) Consider a shower with the following physical parameters. See [Zho06, Section 1.2.1] for a more detailed model. We denote the length of the shower hose, i.e., the distance from the shower tap to the shower head, by l. Let v be the speed of the water in the hose. We assume that the speed of water is constant (stationary flow). The state $x(t)$ is the temperature difference from optimum, i.e., the difference between the (human dependent) optimal temperature and the present temperature at the shower tap.

We now make a somewhat questionable assumption for the system. We model the human being as a linear controller with sensitivity $\alpha > 0$. That is, the change of the controller (shower tap) is assumed to be proportional to the temperature difference of the water coming out of the shower head. The model is described by

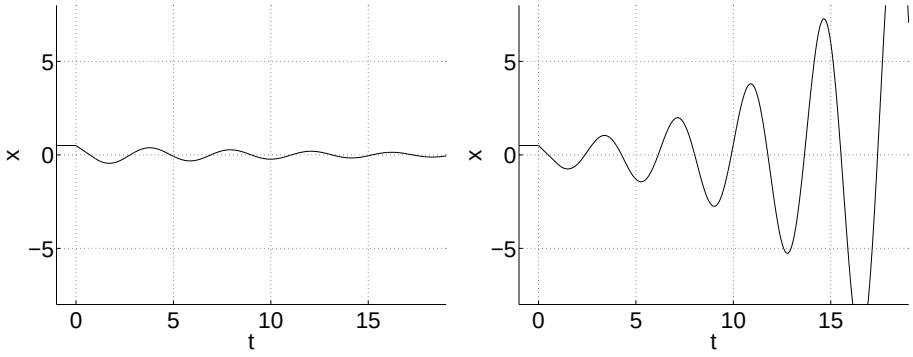
$$\dot{x}(t) = -\alpha x(t-h), \tag{1.1}$$

where $h = l/v$. The solution of this DDE for some choices of the parameters is given in Figure 1.1.

For obvious reasons, you do not want the water temperature to behave like the temperature of the shower in Figure 1.1b. This *unstable* behavior is unwanted in most applications, which is why we will discuss stability conditions for DDEs in Chapter 3. In fact, the DDE (1.1) is stable if and only if

$$h\alpha = \frac{l\alpha}{v} < \frac{\pi}{2}.$$

This is a special case of a well known formula (3.30) given in Chapter 3. It is clear from the exact condition that (as expected) long shower hoses (large l), low water pressure (small v) and sensitive human beings (large α) are the reasons why there are unstable showers.



(a) $\alpha = 1.4$, $h = 1$ (b) $\alpha = 2$, $h = 1$

Figure 1.1: The solution of the hot shower problem

To analyze the stability of (linear homogeneous) ODEs we typically look for solutions with the ansatz $x(t) = x_0 e^{st}$. For the DDE (1.1) this yields the characteristic equation

$$0 = -s - \alpha e^{-sh}. \tag{1.2}$$

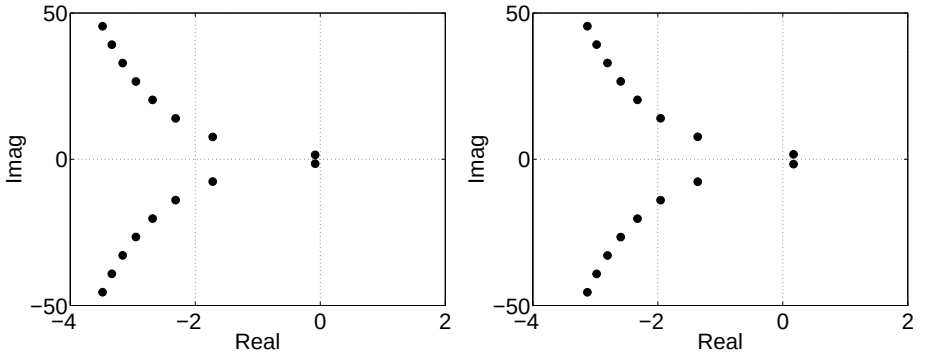
If all solutions of (1.2), i.e., eigenvalues of (1.1), lie in the complex open left half plane, then the system is stable, cf. Figure 1.2.

Similar to ODEs, the solutions of the characteristic equation of a DDE can be used to analyze the stability. Several other properties can also be characterized with the solutions of the characteristic equation, which is why we discuss ways to numerically compute some solutions of the characteristic equation, also called the eigenvalues of the DDE, for *large-scale* DDEs in Chapter 2.

Not even the scalar problem (1.2) can be solved explicitly using the normal elementary functions. However, if we use the logarithmic type function W_k defined as the inverse of $z \mapsto ze^z$, $z \in \mathbb{C}$ we can express the solution of (1.2) explicitly,

$$s = \frac{1}{h} W_k(-h\alpha).$$

Here $k \in \mathbb{Z}$ is the branch index. This function (which is not so widely known) is called Lambert W in the literature. Even though it is not an element of the set of elementary functions, it is available in mathematical software and useful in some



(a) $\alpha = 1.4$, $h = 1$ (b) $\alpha = 2$, $h = 1$

Figure 1.2: The spectrum of the DDE (1.1)

applications. This function, with a corresponding formula for the eigenvalues of some DDEs, is discussed further in Section 2.2.1.

Most aspects of the stability for scalar single delay DDEs, e.g. the hot shower problem, are nowadays considered solved. The situation is different for DDEs of larger dimension. We consider numerical methods for the eigenvalues of large DDEs (with mostly one delay) in Chapter 2. A parameterization of the delays such that there is a purely imaginary eigenvalue, which is valuable information in a stability analysis, is given in Chapter 3, with some new results for scalar problems with multiple delays, as well as computational methods for larger DDEs.

Standard references for DDEs

The book of Bellman and Cooke [BC63] is still commonly cited in literature related to stability and solutions of DDEs. Fundamental results, such as existence and uniqueness, much in the terminology of functional differential equations are treated in [HV93] as well as [DvGVW95]. More recently, [BZ03] seems to be the first monograph related the numerical solution of DDEs. The large amount of literature on stability of DDEs is surveyed in [Nic01a] and [GKC03].

The two monographs [Zho06] and [BL02] treat several topics important in

control theory (and elsewhere), such as implementation, stabilizing feedback, $\mathcal{H}_\infty$-control, and (of course) robust stability.

Finally, even though it does not belong to the standard references (at least not yet), the recent book by Michiels and Niculescu [MN07b] is a modern overview of many results related to stability. This includes more recent topics, e.g. pseudospectra (for DDEs) and stabilization techniques.

Topics of this thesis

The main problems of this thesis are exact conditions for delay-dependent stability (Chapter 3) and numerical methods to compute the spectrum of the DDE (Chapter 2). We also make some notes on perturbation results for the nonlinear eigenvalue problem (Chapter 4).

We first, in Chapter 2, consider the problem of numerically computing the *eigenvalues* of a DDE, i.e., finding solutions of the *characteristic equation*, here referred to as the *delay eigenvalue problem*

$$\left(-sI + A_0 + A_1 e^{-\tau s}\right) v = 0,\ v \neq 0, \tag{1.3}$$

mostly for DDEs with a single delay. As indicated in Example 1.1, the characteristic equation is (informally) motivated as the result of the insertion of the ansatz solution $x(t) = x_0 e^{st}$. See [DvGVW95, Chapter I] for a formal derivation. Because of the nonlinear term in (1.3), the delay eigenvalue problem belongs to a class of problems referred to as *nonlinear eigenvalue problems*. An important contribution of Chapter 2 is the application of a projection method for nonlinear eigenvalue problems, to the author's knowledge, previously not applied on the delay eigenvalue problem. It turns out that this method is superior to other methods to compute the spectrum of DDEs for the presented examples. In particular, we compare it with methods in the literature, and used in software packages. This includes methods based on discretizations of the *solution operator* and discretizations of the equivalent *boundary value problem formulation* of the DDE, i.e., the *infinitesimal generator*. We discuss discretizations based on, but not limited to, *linear multi-step*, *Runge-Kutta* and *spectral collocation*. We also propose some new interpretations of the discretization schemes as rational approximations of the exponential function or the logarithm.

In some cases the spectrum can be expressed explicitly with the function

Lambert W. This formula is valid for scalar DDEs and single-delay DDEs for which the matrices are simultaneously triangularizable. The treated formula is expressed using a matrix version of the Lambert W function. Some works in the literature state this formula in incorrect generality. We state some conditions on the system matrices such that the formula holds. If the system matrices are simultaneously triangularizable the formula holds. We show by counter-example that it does not hold in general.

Chapter 3 deals with exact stability conditions of the DDE. We find exact conditions on the delays such that the DDE has a purely imaginary eigenvalue. All those combinations of the delays such that there is a purely imaginary eigenvalue (called *critical delays*) are parameterized. That is, we construct a mapping consisting of computable expressions, from a simple mathematical object (here $([-\pi, \pi]^{m-1}, \mathbb{Z}^m)$) onto the set of all subsets of the critical delays. For DDEs of arbitrary dimension we give a mapping containing an eigenvalue problem. For the scalar case, we find an explicit expression containing only trigonometric functions. For non-scalar problems, an evaluation of the map consists of solving a *quadratic eigenvalue problem* of squared dimension. The constructed eigenvalue problem is large, even for DDEs of moderate size. For that reason, we present some notes on how the computational cost for one evaluation of the map can be reduced. In particular, we show that the matrix-vector product corresponding to the *companion linearization* of the quadratic eigenvalue problem can be evaluated by solving a *Lyapunov equation*. Most of the results in the chapter on critical delays are derived for retarded DDEs as well as neutral DDEs with an arbitrary number of delays. We also propose a new interpretation of a class of methods for delay-dependent stability, known as *matrix pencil methods*. The new interpretation is based on a similarity with two-parameter eigenvalue problems. We show that the polynomial eigenvalue problems in the matrix pencil methods are the generalized eigenvalue forms of a generalization of the two-parameter eigenvalue problem.

The last chapter, i.e., Chapter 4, is about perturbation results for nonlinear eigenvalue problems. We present some notes on results in eigenvalue perturbation theory which can be generalized to nonlinear eigenvalue problems. A sensitivity formula for the movement of the eigenvalues extends nicely to nonlinear eigenvalue problems. We introduce a fixed point form for the nonlinear eigenvalue problem, and show that some methods in the literature can be interpreted as set-valued fixed point iterations. The convergence order of these types of iterations

can be determined from an expression containing the left and right eigenvectors. We identify that the nonlinear eigenvalue problem belongs to the class of *set-valued fixed point problems* in *fixed point theory*. By using some results available in this field we generalize parts of an important theorem in perturbation theory referred to as *the Bauer-Fike theorem*.

Chapter 2

Computing the spectrum

2.1 Introduction

In this chapter, we consider linear delay-differential equations with a single delay, defined by

$$\Sigma = \begin{cases} \dot{x}(t) = A_0 x(t) + A_1 x(t-\tau),\ t \geq 0 \\ x(t) = \varphi(t),\ t \in [-\tau, 0], \end{cases} \tag{2.1}$$

where $A_0, A_1 \in \mathbb{R}^{n\times n}$, $\tau > 0$ and an initial condition φ, typically assumed to be continuous and bounded.

In the analysis of ordinary differential equations (ODEs) and initial value problems (IVPs), the characteristic equation and the eigenvalues are often used to establish properties of the problem without actually solving it. Similarly, many properties of the DDE (2.1) can be expressed with, or determined from, the characteristic equation and the eigenvalues of the DDE without solving it. The topic of this chapter is numerical methods for the *spectrum* of the DDE which, in our context, is defined as the solution set of the characteristic equation in the following way.

Definition 2.1 *For the DDE* (2.1) *we call:*

i) the equation

$$\det\left(-sI + A_0 + A_1 e^{-s\tau}\right) = 0, \tag{2.2}$$

the characteristic equation *of* (2.1)*;*

ii) a solution $s \in \mathbb{C}$ *to the characteristic equation* (2.2) *an* eigenvalue*;*

iii) the set of all solutions of (2.2) *the* spectrum *of the DDE* (2.1) *and denote it by*

$$\sigma(\Sigma) := \left\{ s \in \mathbb{C} : \det\left(-sI + A_0 + A_1 e^{-\tau s}\right) = 0 \right\};$$

iv) a vector $v \in \mathbb{C}^n \backslash \{0\}$ *corresponding to the eigenvalue* $s \in \sigma(\Sigma)$*, an* eigenvector *corresponding to* s *iff*

$$\left(-sI + A_0 + A_1 e^{-\tau s}\right) v = 0,$$

and the pair (s, v) *is called an eigenpair.*

Moreover, the term delay eigenvalue problem *(DEP), will be used to refer to the problem of finding eigenpairs of a DDE.*

This is a true generalization of the eigenvalues and the characteristic equation of a matrix since if $\tau = 0$ or $A_1 = 0$ then the DDE reduces to an ODE and (2.2) coincides with the characteristic equation of a matrix. Unlike the delay-free case, the characteristic equation generally contains an exponential term. This exponential term (and more generally the characteristic equation) is, in the literature, commonly motivated by looking for non-trivial solutions using the exponential ansatz $x(t) = e^{st}v$. If we insert the exponential ansatz into (2.1) we arrive at the characteristic equation. See e.g. [HV93] or [MN07b] for more formal settings.

Note that, the use of the term *spectrum* in Definition 2.1 is consistent with the common use of the term. The spectrum is normally a property of an operator or a matrix. The terminology of eigenvalues and spectrum of a DDE is indeed consistent, since the DDE (2.1) can be stated as a linear operator acting on a Banach space which spectrum is a point-spectrum [HV93]. That is, the continuous part and the residual part of the spectrum of the operator are empty, which implies that the spectrum consists only of the set of eigenvalues of the operator, i.e., consistent with Definition 2.1.

Also note that the spectrum is independent of the initial condition φ. Even though the spectrum can be used in many different settings, it is particularly often used to analyze properties which are independent of the initial condition.

For instance, we saw in the illustrative example in the previous chapter (Example 1.1) that the spectrum can be used to analyze stability, which is a property independent of the initial condition.

For presentational reasons, it turns out to be advantageous to separate computational and numerical methods related to the spectrum into two contexts:

- *Approaches developed specifically for DDEs (Section 2.2)*

  Even if we assume that the system is scalar, i.e., $n = 1$, the characteristic equation can not be solved for s using the standard exponential and trigonometric functions. There are however approaches exploiting properties of the DDE or the characteristic equation such that the problem can be accurately approximated or solved exactly for special cases.

  Two important concepts in this context are the two operators, the *solution operator* and the *infinitesimal generator*. There are methods specifically developed for the eigenvalues of DDEs derived from approximations of the DDE. Most methods are the result of approximations of these operators in ways which make the characteristic equation easier to solve.

- *Approaches stated in the more general context of nonlinear eigenvalue problems (Section 2.3)*

  The problem of finding the eigenpairs of a DDE belongs to the class problems often called *nonlinear eigenvalue problems*. Here, a nonlinear eigenvalue problem is the problem of finding $s \in \mathbb{C}$ and $v \in \mathbb{C}^n \backslash \{0\}$ such that,

  $$T(s)v = 0, \tag{2.3}$$

  for some parameter dependent matrix $T : \mathbb{C} \to \mathbb{C}^{n \times n}$. The nonlinear eigenvalue problem is a very general class of problems, and there are no general methods which guarantee global convergence (to all solutions). There are however a number of methods with good local convergence properties. See [MV04] and [Ruh73] for overviews of numerical methods for nonlinear eigenvalue problems. The situation is somewhat better for some special cases, e.g. when the eigenvalues are real and $T(s)$ is Hermitian. In particular, adapted projection methods have indeed been successfully applied to some nonlinear eigenvalue problems where T is rational and (2.3) has a (so-called) min-max characterization. See [VW82], [Wer72] and [Vos03].

We present new results in both context. In particular, this chapter contains the following main contributions:

- *A clarification of the generality of an explicit formula containing the Lambert W function (Section 2.2.1).*

  The spectrum of scalar single delay DDEs can be expressed explicitly using the inverse of the function ze^z, referred to as the Lambert W function. One contribution of this chapter is the clarification of the range of applicability of this formula for the delay-eigenvalue problem. The expression is formed by defining a matrix version of Lambert W. The most general sufficient condition for which the formula holds is found to be simultaneously triangularizability of A_0 and A_1.

- *A unified interpretation for the methods based on discretization of the solution operator and the infinitesimal generator in terms of rational approximation (Section 2.2.2 and Section 2.2.3).*

  We propose some new interpretations of current methods for the delay eigenvalue problem. In particular, we point out that several methods in the literature correspond to rational approximations or interpolations. This holds for instance for the methods based on *solution operator discretizations* (SOD). We point out that the SOD-method in the software package DDE-BIFTOOL is a *companion linearization* of a high-order rational eigenvalue problem. It turns out that this rational eigenvalue problem is a Padé approximation of the logarithm. We also discuss the methods based on discretizations of the PDE-formulation of the DDE, which appears to be superior for the numerical examples.

- *An application of a projection method to a delay eigenvalue problem (Section 2.3).*

  The direct application of projection methods to the delay eigenvalue problems seems to be new. For this reason, it is worthwhile to clarify how projection methods can be adapted for this problem (Section 2.3.2). We adapt a subspace acceleration of the vector iteration, *residual inverse iteration* (RII) suggested by Neumaier in [Neu85]. This is an Arnoldi-type projection method and has been used by Voss in [Vos04a] for several other nonlinear eigenvalue problems, e.g., rational eigenvalue problems. In this projection method a small projected nonlinear eigenvalue problem must be

solved in each expansion step. The solution to the projected problem is used in the expansion, which is inspired by one step of the *residual inverse iteration*. We adapt it and apply it to a delay eigenvalue problem.

- *Numerical results for two examples indicating that the projection method is advantageous for large sparse problems (Section 2.4).*

  The methods are applied to two examples in Section 2.4. In the first example with tridiagonal matrices of dimension $n = 10^6$, the projection method finds 12 real eigenvalues in a couple of minutes, whereas the software package DDE-BIFTOOL and other proposed methods breaks down (runs out of memory) for $n = 10^3$. The projection method can solve the second example with random (sparse) matrices for $n = 10^4$. This is also considerably better than the other tested methods.

Note that, to the author's knowledge, delay eigenvalue problems of the size considered here have previously not been solved. This is indicated by the difference in meaning of the term large-scale in the context of DDEs and nonlinear eigenvalue problems. For instance, adapted SOD-methods for delay eigenvalue problems of order 131 (which are called large-scale) are constructed in [VGR04, Section III.B], whereas the nonlinear eigenvalue problems (stemming from modeling of a quantum dot) solved in [Vos06a] are of dimension 10^7 (12 million). The comparison is not entirely fair as the problem in [VGR04] is a bifurcation problem, and hence several eigenvalue problems of this size must be solved. It however still indicates that problems of *large scale* (in the meaning in [Vos04a]) have not been extensively treated in the literature on DDEs.

2.2 Methods for DDEs

If the DDE is scalar or the matrices A_0 and A_1 are simultaneously triangularizable, then $\sigma(\Sigma)$ can be expressed in an explicit way. This is shown in Section 2.2.1. Apart from this special case, the eigenvalues and eigenvectors, i.e., $s \in \mathbb{C}$, $v \in \mathbb{C}^n \backslash \{0\}$ such that

$$(-sI + A_0 + A_1 e^{-\tau s})v = 0 \tag{2.4}$$

can in general not be computed or expressed exactly with elementary or simple operations. In order to numerically solve (2.4), it is natural to consider some form of approximation. The type of approaches we will address next, are based on approximations of (2.4) with an equation which is easier to solve.

Note that the presentation we use here is somewhat different from the typical presentation of similar results in the literature. A lot of literature discuss different types of approximations of the DDE (2.1), whereas the approach we take directly involves aproximations of the characteristic equation (2.2). This allows us to interpet the approximations in a more unified manner.

The problem of determining $s \in \mathbb{C}$ and $v \in \mathbb{C}^n \backslash \{0\}$ such that,

$$\sum_{k=0}^{m} B_k s^k v = 0, \tag{2.5}$$

for $B_0, \ldots, B_m \in \mathbb{C}^{n \times n}$ is called a *polynomial eigenvalue problem*. The polynomial eigenvalue problem can be solved for problems of moderate dimension and order using standard techniques, e.g., companion linearization. Companion linearization is extensively used in this thesis and is further discussed in Appendix A.1. In this section we will show that several methods for the delay eigenvalue problem (2.4) are polynomial or rational approximations, such that the resulting approximation is a polynomial eigenvalue problem (2.5). Obviously, if we approximate $e^{-\tau s}$ with a rational function, we can rewrite the resulting approximation (which is a rational eigenvalue problem) into a polynomial eigenvalue problem (2.5) by multiplying both sides with the denominator. We illustrate this idea with a very crude approximation. We know from the definition of the exponential function that $e^{-\tau s} = \lim_{N \to \infty} (1 + \tau s/N)^{-N}$. Clearly, $e^{-\tau s} \approx S_N(s) := (1 + \tau s/N)^{-N}$ should be an accurate rational approximation for sufficiently large N. Replace $e^{-\tau s}$ in (2.4) with this approximation and multiply by $(1+\tau s/N)^N$. The resulting equation is a polynomial eigenvalue problem

of the type (2.5) with $m = N + 1$. It turns out that this very crude approximation is equivalent to the approximation presented in [BM00], which belongs to a class of methods normally motivated through the semi-discretization of the PDE-formulation of the DDE. This branch of methods will be referred to as IGD (infinitesimal generator discretization). We introduce IGD-methods in Section 2.2.3.

There are two main branches of methods for the delay eigenvalue problem. As mentioned, the methods of one branch are based on the discretization of the infinitesimal generator. The other main branch of methods are motivated by a discretization of the so-called *solution operator*. The methods based on solution-operator discretizations will be referred to as SOD-methods. Several SOD-methods can also be interpreted as rational approximations. We motivated above that some IGD based methods could be interpreted as rational approximations of the exponential term. We will show that some SOD based methods can be interpreted as the approximation of the logarithm in a transformed characteristic equation. In the literature, SOD-methods are derived from a discretization of the solution operator. In particular we will show that the subclass of SOD based methods used in the software package DDE-BIFTOOL[ELR02] are in fact different Padé approximations of $\ln(\mu)$ where μ is a substitution variable $\mu = e^{-hs}$.

It is not surprising that there are two branches of methods for approximations of the spectrum of DDEs. A DDE is indeed a mixture of a difference equation and a differential equation, and the two approaches correspond to the approximation of the differential-term by a difference (Section 2.2.2) and the difference term by a differential expression (Section 2.2.3).

There are other numerical methods to find eigenvalues of DDEs. For instance, the method *quasi-polynomial mapping based rootfinder* (QPMR) [VZ06, Vyh03] is designed for the delay eigenvalue problems. The general idea of QPMR is to separate real and imaginary parts of the characteristic equation $\det(T(s)) = 0$, and to determine the intersection of the two level-set curves. The method uses the coefficients in the characteristic equation. It is well known that the eigenvalues may be very sensitive with respect to perturbations in the coefficients, which may cause serious computational difficulties, as any numerical method will round any intermediate result. This indicates that QPMR may exhibit *numerically unstability* for many problems. Even though this method is sufficiently accurate

to study the general behaviour of eigenvalues for DDEs for many interesting cases, we will not include it in the comparisons done here, as we wish to study eigenvalue problems of large dimension where numerical stability is particularly important.

2.2.1 Scalar or simultaneously triangularizable DDEs: Lambert *W*

The Lambert W function, denoted $W_k(z)$, is the logarithmic type function defined as the multivalued inverse of the complex function $f(w) = we^w$. Here k denotes one of (countably) infinite number of branches.

The Lambert W has been used in a variety of applications (see [CCH+96]) to find explicit formulas for nonlinear equations which were previously mostly solved numerically. In many modern mathematical software packages such as Matlab[1], Maple and Mathematica[2], Lambert W is easily available. Because of its availability in software and the fair amount of applications, some argue that this function should be added to the set of elementary mathematical functions [Hay05]. Throughout this work we will make use of this function as it allows a compact notation.

For the scalar single-delay DDEs

$$\dot{x}(t) = ax(t) + bx(t-\tau),\ a, b \in \mathbb{R}$$

the spectrum can be expressed explicitly using Lambert W, by the well known formula

$$\sigma(\Sigma) = \bigcup_{k\in\mathbb{Z}} \left(\frac{1}{\tau} W_k(\tau b e^{-a\tau}) + a \right), \tag{2.6}$$

where W_k is branch k of Lambert W. In this section we investigate one way to generalize formula (2.6) to some multidimensional time-delay system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bx(t-\tau),\ t \geq 0 \\ x(t) = \varphi(t),\ t \in [-\tau, 0], \end{cases} \tag{2.7}$$

[1]The current standard implementation of Lambert W in Matlab is based on the symbolic toolbox. There is an alternative implementation based on Halley's iteration by Pascal Getreuer without this dependence.

[2]Lambert W is called `ProductLog` in Mathematica

by defining a matrix-version of Lambert W. Note that, we have, for notational purposes, denoted the system matrices A and B, but still denote the set of eigenvalues with $\sigma(\Sigma) := \{s \in \mathbb{C} : s \in \sigma(A + Be^{-\tau s})\}$.

In [AU03] and the derivative works [YU06] and [YUN06] such a generalization for multidimensional systems was attempted. Unfortunately, the result in [AU03] does not hold in the stated generality. We here give sufficient conditions on the system matrices for the formula in [AU03] to hold. As the weakest sufficient condition, we obtain simultaneous triangularizability of the matrices A and B. Similar observations have been made recently in [SM06], where basically the same spectral results are obtained without explicit use of a matrix version of the Lambert W function. Here we establish these results for the representation in [AU03]. Moreover, we present an explicit counterexample, which proves that in general, the formula may be wrong. This is important, since some of the results of [AU03] have been cited in articles in a wide variety if fields, e.g. in [HC05], [WC05], [HS05], [CM02b], [CM02a], [KN05], [Bam07], [HC06], [Pit05],[WD07],[LJH06], [GCS08] and [ACS05]. Even if most of the conclusions drawn in these papers still seem to be valid, since mainly the scalar case is considered, it is worthwhile to clarify the range of applicability of the formula.

Large parts of these results were published in a joint work with Tobias Damm in [JD07]. Independent and similar observations regarding the accuracy of [AU03] were published in the discussion article [Zaf07] together with a relevant application to a problem from machine tool chatter.

The Lambert W function

For scalar arguments z, the *Lambert W function* is defined as the (multivalued) inverse of the function $z \mapsto ze^z$. It has a countably infinite number of branches

$$W_k(z) \in \{w \in \mathbb{C} : z = we^w\}\,, \quad k \in \mathbb{Z}\,,$$

which can be defined by the branch-cuts in [JHC96] and [CCH$^+$96]. Note that unlike some fundamental functions, say the complex-valued logarithm, numbering of the branches is not obvious and not periodic (as is the case for the logarithm). Each of the branches is locally analytic in all points but $z = -1/e$. Hence, we may define a matrix-version of the Lambert W function in a standardized way, given e.g. in [HJ91] or [Hig06]. We first define Lambert W for matrices in Jordan

canonical form, i.e.,

$$J = \operatorname{diag}(J_{n_1}(\lambda_1), J_{n_2}(\lambda_2), \ldots, J_{n_s}(\lambda_s)) \ ,$$

where $J_n(\lambda)$ is the n-by-n Jordan block belonging to eigenvalue λ with multiplicity n. Then

$$W_k(J) = \operatorname{diag}(W_{k_1}(J_{n_1}(\lambda_1)), \ldots, W_{k_s}(J_{n_s}(\lambda_s))) \ .$$

Note that we are allowed to pick a different branch for each Jordan block. If J has s Jordan blocks and the index set for the branches of the scalar Lambert W function is $\mathbb{Z}$, then the index set for the branches of $W_k(J)$ is $\mathbb{Z}^s$. For Jordan blocks of dimension 1, i.e., single eigenvalues, we can use the scalar Lambert W function. For all other cases, we define the Lambert W function (for a fixed branch) of a Jordan block by the standard definition of matrix functions (e.g. see [HJ91, Eq. (6.18)]), i.e.,

$$W_k(J_n(\lambda)) = \begin{pmatrix} W_k(\lambda) & W_k'(\lambda) & \cdots & \frac{1}{(n-1)!}W_k^{(n-1)}(\lambda) \\ 0 & W_k(\lambda) & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & W_k(\lambda) \end{pmatrix} .$$

We complete the definition of Lambert W for matrices, by noting that all matrices can be brought to Jordan canonical form by a similarity transformation $A = SJS^{-1}$. Thus we may set $W_k(A) = SW_k(J)S^{-1}$, where for the principal branch $k = 0$ we from now on tacitly assume that $-e^{-1}$ is not an eigenvalue corresponding to a Jordan-block of dimension larger than 1, i.e.,

$$\operatorname{rank}(A + e^{-1}I) = \operatorname{rank}(A + e^{-1}I)^2 \ . \tag{2.8}$$

Remark 2.2 The limitation (2.8) lessens the elegance of the matrix Lambert W function slightly. This point was brought to our knowledge by Robert Corless.

Example 2.3 *We illustrate the definition of the Lambert W function for a* 2×2*-Jordan block.*
Let $J = \begin{bmatrix} z & 1 \\ 0 & z \end{bmatrix}$, *then* $W_k(J) = \begin{bmatrix} W_k(z) & W_k'(z) \\ 0 & W_k(z) \end{bmatrix}$. *We verify that indeed*

$J = W_k(J)e^{W_k(J)}$. *To this end we note that by differentiating the equation* $z = W_k(z)e^{W_k(z)}$, *we obtain* $1 = W_k'(z)e^{W_k(z)} + W_k'(z)z$. *Thus we have*

$$\begin{aligned} W_k(J)e^{W_k(J)} &= e^{W_k(z)} \begin{bmatrix} W_k(z) & W_k'(z) \\ 0 & W_k(z) \end{bmatrix} \begin{bmatrix} 1 & W_k'(z) \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} z & zW_k'(z) + e^{W_k(z)}W_k'(z) \\ 0 & z \end{bmatrix} = J \, . \end{aligned}$$

Conditions for the Lambert *W* formula

With the help of the Lambert W function we can easily express the spectrum of triangular systems.

Lemma 2.4 *If* A *and* B *are both upper or both lower triangular matrices, then*

$$\sigma(\Sigma) = \bigcup_k \sigma\left(\frac{1}{\tau}W_k(B\tau e^{-A\tau}) + A\right) \, . \tag{2.9}$$

Proof: We exploit the fact that the determinant of a triangular matrix is the product of the diagonal elements. The characteristic equation is hence

$$0 = \det\left(-sI + A + Be^{-s\tau}\right) = \prod_j (-s + a_{jj} + b_{jj}e^{-s\tau}).$$

Clearly, $-s + a_{jj} + b_{jj}e^{-s\tau} = 0$ for some j, if and only if s is an eigenvalue. It follows that

$$(s - a_{jj})\tau e^{(s-a_{jj})\tau} = b_{jj}\tau e^{-a_{jj}\tau},$$

which, for any branch W_k results in

$$s = \tfrac{1}{\tau}W_k(b_{jj}\tau e^{-a_{jj}\tau}) + a_{jj}.$$

The expression holds for all choices j, hence

$$s \in \sigma(\Sigma) = \bigcup_{k,j\in\mathbb{Z}} \tfrac{1}{\tau}W_k(b_{jj}\tau e^{-a_{jj}\tau}) + a_{jj} = \bigcup_{k\in\mathbb{Z}} \sigma\left(\tfrac{1}{\tau}W_k(B\tau e^{-A\tau}) + A\right) \, ,$$

completing the proof. □

Lemma 2.4 can easily be extended to the case where A and B are simultaneously triangularizable in the following sense.

Definition 2.5 *The matrix pair A, $B \in \mathbb{C}^{n\times n}$ is called* simultaneously triangularizable *if there is a regular $S \in \mathbb{C}^{n\times n}$ and upper triangular matrices T_A and T_B such that*

$$A = S^{-1}T_A S \quad \text{and} \quad B = S^{-1}T_B S \; .$$

Assuming simultaneous triangularizability, we can introduce new variables $\xi = Sx$, such that system (2.7) can be written as a cascade of inhomogeneous scalar equations

$$\dot{\xi}_j(t) = \alpha_j \xi_j(t) + \beta_j \xi_j(t-h) + \gamma_j(t) \; ,$$

where γ_j is a linear combination of the functions $\xi_1, \ldots, \xi_{j-1}$. The spectrum of the whole system is the union of the spectra of these scalar equations. We thus obtain the most general case for the formula to hold.

Theorem 2.6 *If A and B are simultaneously triangularizable, then (2.9) holds.*

Proof: The characteristic equation is invariant under simultaneous similarity transformation i.e.,

$$\det(-sI + A + Be^{-s\tau}) = \det(-sI + T_A + T_B e^{-s\tau}) \; .$$

Moreover, the exponentiation operator and Lambert W commute with similarity transformation, i.e.,

$$W(S^{-1}CS) = S^{-1}W(C)S \; .$$

This implies that (2.9) is invariant under simultaneous similarity transformation of A and B. Hence we can assume without loss of generality that A and B are both upper triangular and apply Lemma 2.4. □

We mention some interesting special cases.

Corollary 2.7 *If A and B commute, then (2.9) holds.*

Proof: This follows from Theorem 2.6 and the fact that commutativity implies simultaneous triangularizability (cf. [RR00]). □

This result for $\tau = 1$ is also stated (without proof) in [CCH+96]. It implies that (2.9) also holds in the *pure delay case.*

Corollary 2.8 *If $A = 0$ then $\sigma(\Sigma) = \bigcup_k \sigma(\frac{1}{\tau} W_k(\tau B))$.*

Proof: The matrices B and 0 commute, which allows us to apply the previous corollary. □

Finally we note two partial results related to the controllability of the matrices as defined in [Son98].

Lemma 2.9 *Assume that the pair (A, B) is not controllable, i.e. the matrix*

$$[B, AB, \ldots, A^{n-1}B]$$

has rank less than n, and $\sigma_u(A)$ denotes the corresponding set of uncontrollable eigenvalues of A. Then

$$\sigma_u(A) \subset \sigma(\Sigma) \cap \bigcup_k \sigma\left(\frac{1}{\tau} W_k(B\tau e^{-A\tau}) + A\right) .$$

Proof: By the Kalman decomposition (e.g. [Son98, Lemma 3.3.3]) there exists a nonsingular S so that

$$S^{-1}AS = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} , \quad S^{-1}BS = \begin{bmatrix} B_{11} & B_{12} \\ 0 & 0 \end{bmatrix} ,$$

where $\sigma(A_{22}) = \sigma_u(A)$. We can assume A and B to be in this form. Hence $\sigma(A_{22}) \subset \sigma(\Sigma)$. Now we consider $\sigma(\frac{1}{\tau} W_k(B\tau e^{-A\tau}) + A)$. Here $B\tau e^{-A\tau}$ has the form $\begin{bmatrix} X & Y \\ 0 & 0 \end{bmatrix}$, and $W = \begin{bmatrix} W_k(X) & Ye^{-W_k(X)} \\ 0 & 0 \end{bmatrix}$ satisfies $We^W = B\tau e^{-A\tau}$. Thus $\sigma(A_{22}) \subset \sigma\left(\frac{1}{\tau} W_k(B\tau e^{-A\tau}) + A\right)$ for some branch of the Lambert W function. □

A counter-example

To demonstrate that formula (2.9) is not applicable to arbitrary DDEs, we pick the following pair of matrices which are not simultaneously triangularizable and not commuting:

$$A = \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix} , \quad B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} ,$$

for some $\alpha \in \mathbb{R}$, $\alpha > 0$. We now find an explicit expression for the eigenvalues. The characteristic equation is given by

$$0 = \det\left(-sI + A + Be^{-s\tau}\right) = s^2 - \alpha e^{-s\tau} . \tag{2.10}$$

Eigenvalues s are thus characterized by

$$\alpha = s^2 e^{s\tau} \quad \Longleftrightarrow \quad \pm\frac{1}{2}\tau\sqrt{\alpha} = \frac{1}{2}s\tau e^{\frac{1}{2}s\tau} \ . \tag{2.11}$$

In particular, $s_0 = \frac{2}{\tau}W_0\left(\pm\frac{1}{2}\tau\sqrt{\alpha}\right)$ is an eigenvalue, where W_0 denotes the principal branch of the Lambert W function. The example becomes explicitly tractable, if we pick $\tau = 1$, $\alpha = \pi^2$ and make use of the fact that $W_0(-\frac{1}{2}\pi) = \frac{1}{2}\pi i$. Hence we obtain $s_0 = \pi i$.

By formula (2.9) we would have

$$\sigma(\Sigma) = \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right) \ , \tag{2.12}$$

where again $\tau = 1$. It is clear from Figure 2.1 that this expression is not consistent with (2.11).

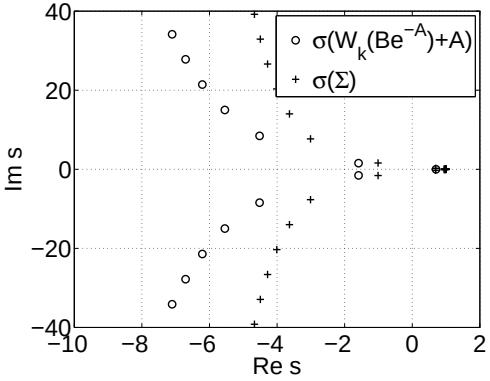


Figure 2.1: Counter-example from Section 2.2.1 with $\tau = \alpha = 1$

To prove this strictly, we first find an s such that $s \in \sigma(\Sigma)$, but not $s \in \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$, and hence prove that $\sigma(\Sigma) \not\subset \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$. Secondly, we find an s such $s \in \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$ but not $s \in \sigma(\Sigma)$, and prove that $\sigma(\Sigma) \not\supset \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$.

We note that $Be^{-A} = \begin{pmatrix} -\alpha & 1 \\ 0 & 0 \end{pmatrix}$, and

$$W_k(Be^{-A}) = \begin{pmatrix} W_k(-\alpha) & -\frac{1}{\alpha}W_k(-\alpha) \\ 0 & 0 \end{pmatrix}.$$

By (2.12), eigenvalues would be characterized via

$$0 = s^2 - sW_k(-\alpha) + W_k(-\alpha)\ , \tag{2.13}$$

or more explicitly,

$$s = \frac{W_k(-\alpha) \pm \sqrt{W_k(-\alpha)^2 - 4W_k(-\alpha)}}{2}\ . \tag{2.14}$$

In particular, for $\alpha = \pi^2$ and some $k \in \mathbb{Z}$, the eigenvalue $s_0 = \pi i$ would have to satisfy (2.13). Thus

$$\begin{aligned} 0 &= (i\pi)^2 - (i\pi)W_k(-\pi^2) + W_k(-\pi^2) \\ &= W_k(-\pi^2)(1 - i\pi) - \pi^2\ . \end{aligned}$$

Hence we conclude $W_k(-\pi^2) = \frac{\pi^2}{1-i\pi}$. This is not fulfilled for any branch k since

$$-\pi^2 = \tfrac{\pi^2}{1-i\pi} e^{\frac{\pi^2}{1-i\pi}} \quad \Longleftrightarrow \quad i\pi - 1 = e^{\frac{\pi^2}{1-i\pi}}\ .$$

Taking absolute values, we get $\sqrt{\pi^2+1} = e^{\frac{\pi^2}{1+\pi^2}}$ which contradicts $\pi > e$. Hence $\sigma(\Sigma) \not\subset \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$.

Note that, vice versa, we can also produce an explicit example, where $\sigma(\Sigma) \not\supset \bigcup_k \sigma\left(W_k(Be^{-A}) + A\right)$. Let $\alpha = \frac{1}{2}\pi$. For the principal branch of W equation (2.14) reduces to

$$s = \frac{i\pi \pm \sqrt{-\pi^2 - 8\pi i}}{4}\ .$$

It remains to show that s does not always satisfy the characteristic equation $s^2 = \frac{\pi}{2}e^{-s}$ from (2.10). Setting $a + bi = \pm\sqrt{-\pi^2 - 8\pi i}$ with $a > 0$ we find $ab = -4\pi$, whence $b < 0$, and $a^2 - b^2 = -\pi^2$, whence $b < -\pi$. Moreover $b > -2\pi$, since otherwise $a^2 = b^2 - \pi^2 \geq 3\pi^2$ and $a^2b^2 \geq 12\pi^4 > 16\pi^2$. Thus Re $s > 0$ and $0 >$ Im $s > -\pi/4$, which implies Im $s^2 < 0$, Im $e^{-s} > 0$, i.e. $s^2 \neq \frac{\pi}{2}e^{-s}$.

Actually, the spectra are disjoint (Figure 2.1), which is consistent with Lemma 2.2.1 and the controllability of (A, B).

One may ask, whether our counter-example hinges on the controllability of (A, B). It is, in fact, an immediate consequence of the Kalman decomposition that any non simultaneously triangularizable pair of 2×2 matrices is controllable. We may, however, embed our example in a higher-dimensional uncontrollable system $\tilde{G}$, setting

$$\tilde{A} = \begin{pmatrix} A & 0 \\ 0 & 1 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix},$$

so that

$$\det\left(-sI + \tilde{A} + \tilde{B}e^{-s}\right) = (1-s)\det\left(-sI + A + Be^{-s}\right) ,$$

i.e., $\sigma(\tilde{\Sigma}) = \sigma(\Sigma) \cup \{1\}$, and (for all branches W_k)

$$\begin{aligned} &\det\left(-sI + W_k(\tilde{B}e^{-\tilde{A}}) + \tilde{A}\right) \\ &= (1-s)\det\left(-sI + W_k(Be^{-A}) + A\right) , \end{aligned}$$

i.e., $\sigma\left(W_k(\tilde{B}e^{-\tilde{A}}) + \tilde{A}\right) = \sigma\left(W_k(Be^{-A}) + A\right) \cup \{1\}$. Our conclusions thus hold for this system as well, where in accordance with Lemma 2.2.1 the uncontrollable eigenvalue 1 is contained in $\sigma(\tilde{\Sigma}) \cap \sigma\left(W_k(\tilde{B}e^{-\tilde{A}}) + \tilde{A}\right)$.

2.2.2 Solution operator discretization (SOD)

There are two common ways to express a DDE in terms of operators. The first representation treated next, is based on the *solution operator* (the second approach is discussed in Section 2.2.3). The solution operator is the operator transforming an initial condition φ onto the solution segment at a later time-point specified by a parameter h, in the following sense.

Definition 2.10 (Solution operator) *The* solution operator *of the DDE* (2.1) *is the operator transforming an initial condition ϕ to the solution segment at time-point h. We denote this operator by $T(h) : \mathcal{C}([-\tau, 0], \mathbb{R}^n) \to \mathcal{C}([-\tau, 0], \mathbb{R}^n)$. The solution operator applied to ϕ, i.e., $(T(h)\phi)(\theta) =: \psi(\theta)$, is the solution segment of* (2.1) *with initial condition $\varphi = \phi$ at time-point h. More precisely,*

$$\psi(\theta) := (T(h)\phi)(\theta) = x(h+\theta),\ \theta \in [-\tau, 0],$$

where $x(t)$ is the solution of (2.1) *with initial condition $\varphi = \phi$.*

The solution operator is commonly used in many standard works, e.g., [HV93, Chapter 2], [DvGVW95, Chapter I] and [MN07b]. In these and other works it is common to denote the function segment of the solution to the left of time-point h (with length τ) by x_h. That is, x_h is the function window with view x of width τ at time-point h. Formally, $x_h(\theta) := x(h+\theta)$ for $h \geq 0$ and $-\tau \leq \theta \leq 0$. In this terminology, the definition of the solution operator is $T(h)\varphi = x_h$.

A number of discretizations of the solution operator have been used in the literature. For instance, it is the basis for the eigenvalue computations in the software package DDE-BIFTOOL [Eng00]. Note that in this context, the solution operator is not (despite the name) used to numerically find a solution of the DDE, but constructed in order to approximate the eigenvalues.

We start the discussion by reviewing the typical derivation performed in the literature, i.e., by formulating a (linear) solution operator $T(h)$, approximate the operator with a difference scheme (normally an equidistant *linear multistep method*) from which the eigenvalues of the DDE can be approximated by solving a large eigenvalue problem. After outlining the derivations done in the literature, we present an alternative motivation by noting that the discretization can be equivalently interpreted as a Padé approximation of $\ln(\mu)$ where $\mu = e^{hs}$, for the step-size h. We believe that the alternative derivation is somewhat more natural for the purpose of computing eigenvalues as it does not require the introduction of the solution operator.

The literature in the field of delay-differential equations has a strong foundation in *functional analysis*. It has clearly colored the results and terminology in the field. This is clear from the presentation in (or just the titles of) two of the standard references, the book by Diekmann, et al. [DvGVW95] and the book of Hale and Verduyn Lunel [HV93] where delay-differential equations are treated as functional differential equations. Since functional analysis is not a main topic of this work, we wish to do a fairly self-contained presentation of current numerical methods by only introducing the concepts from functional analysis necessary for the derivation. The solution operator and the infinitesimal generator in the next section (Section 2.2.3) are necessary concepts.

It turns out that the solution operator can be expressed in an explicit way if we assume that $h \leq \tau$. In this case, one part of the operator $T(t)$ is a shift, e.g. large parts of x_t and $x_{t+\Delta}$ are equal for small Δ, cf. Figure 2.3. The other part of the operator can be expressed by the differential part of (2.1). We have just

provided a motivation for the explicit expression of the solution operator which is a combination of a shift and an *ordinary differential equation* (ODE),

$$(T(h)\varphi)(\theta) = \begin{cases} \psi(\theta) = \varphi(\theta + h) & \theta \leq -h \\ \text{Solution of } \dot{\psi}(\theta) = A_0\psi(\theta) + A_1\varphi(\theta + h - \tau) & \theta \geq -h, \end{cases} \tag{2.15}$$

for $h < \tau$. Note that $A_1\varphi(\theta + h - \tau)$, i.e., the second term in the ODE-case, is a previous time-point, which is known, and can be interpreted as an inhomogeneous part of the ODE. The initial condition for the ODE in the second case is such that $T(t)\varphi$ is continuous, i.e., $(T(h)\varphi)(-h) = \varphi(0)$, cf. Figure 2.2. This construction is sometimes referred to the *method of steps*, and gives a natural way to integrate a DDE by (numerically) solving the ODE-part in each step (but in this work we focus on eigenvalues of DDEs and not the integration of DDEs).

We formalize this construction in a theorem.

Theorem 2.11 *Consider the DDE* (2.1) *with the solution operator* $T(h)$ *defined by Definition 2.10. Suppose* $h \leq \tau$*, then for any* $\varphi \in \mathcal{C}([-\tau, 0])$*,*

$$(T(h)\varphi)(\theta) = \begin{cases} \psi(\theta) = \varphi(\theta + h) & \theta \in [-\tau, -h] \\ \textit{Solution of } \dot{\psi}(\theta) = A_0\psi(\theta) + A_1\varphi(\theta + h - \tau) & \theta \in [-h, 0]. \end{cases} \tag{2.16}$$

Proof: First, suppose $\theta \in [-\tau, -h]$. Then, the evaluation of $\psi(\theta) := x(h + \theta)$ is always the initial condition in the definition of the DDE (2.1) since $h + \theta \leq 0$. Hence, $x(h + \theta) = \varphi(h + \theta)$. This proves the first case in (2.16).

Now, suppose $\theta \in [-h, 0]$. Since $\psi(\theta) := x(h + \theta)$ and $h + \theta \geq 0$ we use the first case in the definition of the DDE (2.1) to evaluate $x(h + \theta)$, i.e.,

$$\dot{\psi}(\theta) = A_0\psi(\theta) + A_1x(h + \theta - \tau). \tag{2.17}$$

Now note that since $h \leq \tau$, $h+\theta-\tau$ is non-negative and the evaluation $x(h+\theta-\tau)$ is the second case (the initial condition) in the definition of the DDE (2.1), i.e., $x(h + \theta - \tau) = \varphi(h + \theta - \tau)$. Also note that this case, i.e., the second case in (2.16) is an inhomogeneous initial value problem where the initial value $\psi(-h)$ is taken from the first case. Existence and uniqueness of a solution ψ of (2.17) for the interval $\theta \in [-h, 0]$ follows from the theorem of Picard-Lindelöf [Lin94]. The proof is completed by noting the fact that the left and the right hand side in (2.16) are both uniquely defined. □

The construction has an important role in theoretical analysis of DDEs. For instance, the uniqueness of the forward solution of DDEs follows from the uniqueness of solutions of ODEs and even if the initial condition is continuous but not differentiable, the next interval is differentiable. More generally, the number of existing derivatives increases with one per time-interval. This is referred to as the *smoothing property* of (retarded) DDEs; cf. illustration in Figure 2.2a.

The operator is clearly linear, and the spectrum is related to the spectrum of the DDE by the following *spectral mapping principle* given in e.g. [DvGVW95, Appendix II, Theorem 4.17]. For any $t > 0$

$$\sigma(\Sigma) = \frac{1}{t}\ln(\sigma(T(t))\backslash\{0\}), \tag{2.18}$$

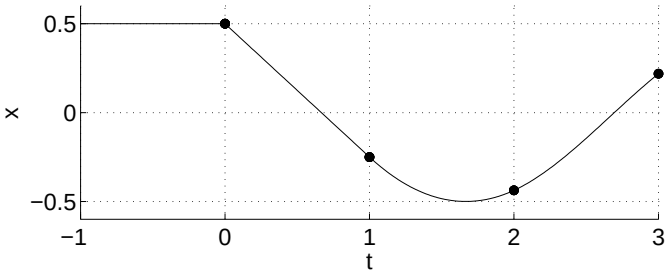
where the logarithm is the set of all branches of the component application on the elements of the set $\sigma(T(t))\backslash\{0\}$. In the following example we illustrate the equivalence by giving an explicit expression $T(t)$ for the special case that $t = \tau$. Using the Lambert W, we can compute the eigenvalues of $T(\tau)$ and confirm that the equivalence holds by comparing the resulting expression with the formula derived in Section 2.2.1.

Example 2.12 *Consider*

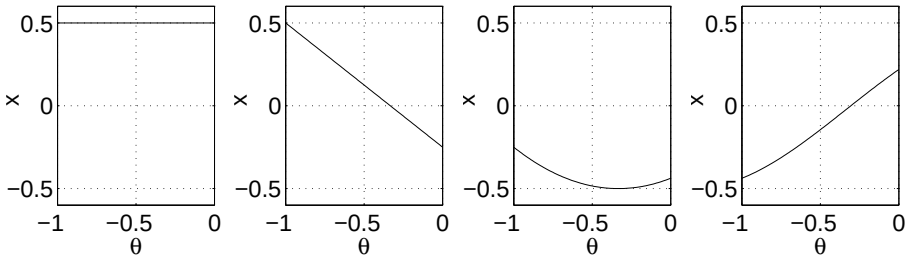
$$\begin{cases} \dot{x}(t) = -\frac{3}{2}x(t-\tau) & t \geq 0 \\ x(t) = \varphi_0(t) & t \in [-\tau, 0] \end{cases} \tag{2.19}$$

As mentioned, we call the function segment of the solution x to the left of some time-point t, x_t. That is, $x_t(\theta) = x(t+\theta)$, $\theta \in [-\tau, 0]$. We now wish to construct the operator $T(t)$ which transforms the initial function condition into the function segment at time point t, i.e., $x_t = T(t)\varphi_0$. It is clear, from (2.15) *and Figure 2.2, that for $t = \tau$ the operator is an integration and takes the particularly easy form $(T(\tau)\varphi)(\theta) = -\frac{3}{2}\int_{-\tau}^{\theta}\varphi(t)\,dt + \varphi(0)$. The result of the application of the solution operator to scalar DDEs with $a_0 = 0$ and simple initial conditions can be computed exactly, e.g., here the solution segments are polynomial with increasing order, for instance if $\tau = 1$,*

$$\begin{aligned} (T(1)\varphi_0)(\theta) &= -\frac{1}{4} - \frac{3}{4}\theta \\ (T(2)\varphi_0)(\theta) &= (T(1)^2\varphi_0)(\theta) = -\frac{7}{16} + \frac{3}{8}\theta + \frac{9}{16}\theta^2 \\ (T(3)\varphi_0)(\theta) &= (T(1)^3\varphi_0)(\theta) = \frac{7}{32} + \frac{21}{32}\theta - \frac{9}{32}\theta^2 - \frac{9}{32}\theta^3. \end{aligned}$$



(a) The solution $x(t)$. The dots • mark discontinuities in the derivatives



(b) $\varphi_0(\theta) = 0.5$ (c) $T(\tau)\varphi_0$ (d) $T(2\tau)\varphi_0 = T(\tau)^2\varphi_0$ (e) $T(3\tau)\varphi_0 = T(\tau)^3\varphi_0$

Figure 2.2: Graphical interpretation of Example 2.12 with initial condition $\varphi_0 = 0.5$, $\tau = 1$.

We now search for eigenvalues of the operator $T(\tau)$, i.e., $\mu \in \mathbb{C}$, and nontrivial functions φ such that

$$\mu\varphi = T(\tau)\varphi.$$

Inserting the definition of T (2.15) for $\tau = h$, yields

$$\mu\varphi'(\theta) = -\frac{3}{2}\varphi(\theta). \tag{2.20}$$

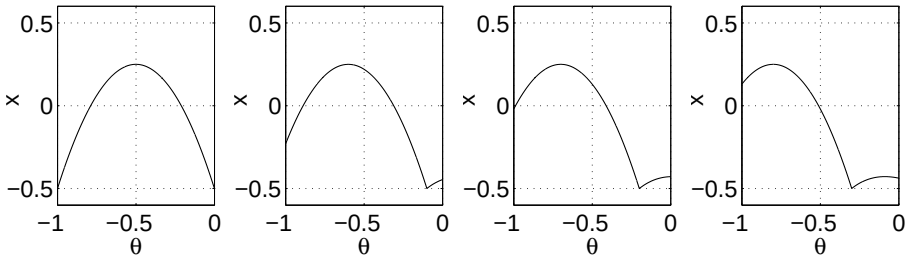
Moreover, we have,

$$\mu\varphi(-\tau) = \varphi(0). \tag{2.21}$$

From (2.20), the eigenfunctions are $\varphi(\theta) = \alpha e^{-\frac{3}{2\mu}\theta}$, where α is a normalization constant. From the second condition (2.21) we have that $\varphi(0) = \alpha = \mu\varphi(-\tau) = \alpha\mu e^{3\tau/2\mu}$, i.e., $\mu = \frac{-3\tau}{2W_k(-3\tau/2)}$ with the corresponding eigenfunction $\varphi(\theta) =$

$e^{W_k(-\frac{3\tau}{2})\theta}$, *i.e.,* $\sigma(T(\tau)) = \{\frac{-3\tau}{2W_k(-3\tau/2)}\}$. *From the spectral mapping principle* (2.18)*, the eigenvalues of the DDE are* $s = \frac{1}{\tau}\ln(\mu) = \frac{1}{\tau}\ln(\frac{-3\tau}{2W_k(-3\tau/2)}) = \frac{1}{\tau}W_k(-\frac{3\tau}{2})$. *This is consistent with the explicit expression from* (2.6).

In the example above we expressed $T(\tau)$ in an exact way. We will now discuss a way to approximate $T(h)$ for typically a small h. This is a main idea in the Matlab package DDE-BIFTOOL [Eng00]. First note that for small h, a large part of $T(h)$ is a shift, cf. Figure 2.3. To be precise, suppose $h < \tau$, then from (2.15) it holds that $(T(h)\varphi)(\theta) = \varphi(\tau + \theta)$ for all $\theta \in (-\tau, -h]$. The general goal is to construct a discretization which preserves the shift property exactly, but approximates the ODE-part in the "new" segment $(-h, 0)$. In DDE-BIFTOOL this is done with a *linear multistep* (LMS) discretization.



(a) $x_0 = \varphi_0$ (b) $x_{0.1} = T(0.1)\varphi_0$ (c) $x_{0.2} = T(0.2)\varphi_0$ (d) $x_{0.3} = T(0.3)\varphi_0$

Figure 2.3: Large parts of $T(h)\varphi_0$ is a shift if h is small. Here applied to Example 2.12.

Suppose the interval $[-\tau, 0]$ is discretized into N nodes, i.e., step-length $h = \tau/(N-1)$. The solution for the discretization of the solution operator $T(h)$ can be written as a shift for all elements but the last, which must be an approximation of the DDE. Note that we pick the step-length h and evaluate the point of evaluation of T equal, i.e., $t = h$.

In a first example we approximate the last segment with a finite difference.

Example 2.13 *We show the typical way the discretization of the solution operator is motivated by treating a simple DDE with a simple discretization. Again,*

consider

$$\begin{cases} \dot{x}(t) = -\frac{3}{2}x(t-1) & t \geq 0 \\ x(t) = \varphi(t) & t \in [-1, 0]. \end{cases}$$

We now discretize the interval $\theta \in [-\tau, 0]$ *with* N *equidistant nodes, where* $h = \tau/(N-1)$*, that is* $\varphi(\theta_j) \approx u_j$*, where* $\theta_j = (j-1)h$*.*

We will construct an approximation $S_N \in \mathbb{R}^{N\times N}$ *of* $T(h)$ *by imposing the shift property, and approximating the new interval with a difference scheme. In the following we denote the element of a vector by using indices, i.e., element* k *of the vector* u *is denoted by* u_k *or* $(u)_k$*. The shift consistency can be guaranteed by choosing* S_N *such that* $(S_N u)_k = u_{k+1}$ *for all* $k = 1, \ldots, N-1$*. We establish a relation for the* N*th component by approximating the DDE. In this first example we approximate the DDE with a forward difference, i.e., here*

$$\frac{(S_N u)_N - (S_N u)_{N-1}}{h} = -\frac{3}{2}(S_N u)_1,$$

or $(S_N u)_N = (S_N u)_{N-1} - \frac{3}{2}h(S_N u)_1$*. The shift and the finite differences define the matrix* S_N*, i.e.,*

$$S_N = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -\frac{3}{2}h & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Since the eigenvalues of S_N *approximate the eigenvalues of* $T(h)$*, the eigenvalues of the DDE are approximated by* $s = \frac{1}{h}\ln\mu$ *where* μ *is an eigenvalues of* S_N*. From the spectral mapping principle,* (2.18)*,*

$$\exp(h\sigma(\Sigma)) = \sigma(T(h)).$$

Note that the complex inverse of the exponential is not unique. However, there is a branch of the logarithm such that if $z \in \sigma(T(h))$ *then*

$$s = \frac{1}{h}\ln(z)$$

implies that $s \in \sigma(\Sigma)$*. In practice, in particular for stability analysis, it is often not necessary to determine the branch since the real part of all branches of the logarithm are equal.*

The idea of the alternative derivation (based on the Padé approximation of the logarithm) which we will address below, is that the matrix S_N is a companion matrix and the eigenvalues μ of S_N are actually the roots of the polynomial $0 = -\mu^N + \mu^{N-1} - \frac{3}{2}h$.

In the example above we used a forward difference scheme to approximate the ODE. A numerous number of other discretizations have been successfully used. We will review some results for the *linear multistep* discretizations and *Runge-Kutta* discretizations. We call these methods SOD(LMS) and SOD(RK) respectively.

Linear multistep, described in any undergraduate text-book on numerical methods for ODEs (e.g. [Lam91]), is a way to numerically solve the initial value problem,

$$\begin{aligned} \psi(a) &= \eta, \\ \psi'(\theta) &= f(\theta, \psi). \end{aligned}$$

Linear multistep is a method specified by the constants $\alpha_0, \ldots, \alpha_k$ and $\beta_0, \ldots, \beta_k$ and an approximation given by the relation

$$\sum_{j=0}^{k} \alpha_j \psi_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}. \tag{2.22}$$

Note that the second part of the solution operator (2.15) is an initial value problem as above, if we let

$$f(\theta, \psi) = A_0 \psi + A_1 \varphi(\theta + h - \tau). \tag{2.23}$$

We will apply the discretization of the solution operator to an equidistant set of grid-points $\theta_j = jh$, $j = -N, \ldots, 0$, where $h = \frac{\tau}{N}$. We denote an approximation at point θ_j by ψ_j, i.e., $\psi(\theta_j) \approx \psi_j$. Analogously, $\varphi(\theta_j) \approx \varphi_j$. For notational convenience we also let

$$f_j = f(\psi_j, \theta_j) = A_0 \psi_j + A_1 \varphi_{j+1-N}. \tag{2.24}$$

We now show a way to give the discretization with step-size h of the solution operator (2.15) $T(h)$ by expressing the discretization of ψ_j, $j = -N, \ldots, 0$ in terms of φ_j, $j = -N, \ldots, 0$.

We have fixed the step-size h equal to the evaluation point of the solution operator $T(h)$. This makes the shift-part of the operator expressible as

$$\psi_j = \varphi_{j+1},\ j = -N, \dots -1.$$

This comes from the fact that only one discretization point θ_0 lies in the ODE-part of the solution operator.

It remains to express ψ_0 in terms of the discretization φ. Here, this is carried out with the LMS-scheme.

We let $n = -k$ in the LMS-scheme (2.22) such that the rightmost point is θ_0. That is,

$$\sum_{j=0}^{k} \alpha_j \psi_{j-k} = h \sum_{j=0}^{k} \beta_j f_{j-k}. \tag{2.25}$$

This equation can be solved for ψ_0 in the following way. We now use the definition of f_j (2.24) and again use the shift property. The shift property can be applied to $\psi_{-k}, \dots, \psi_{-1}$, yielding,

$$\alpha_k \psi_0 + \sum_{j=0}^{k-1} \alpha_j \varphi_{j-k+1} = h\beta_k A_0 \psi_0 + h \left(\sum_{j=0}^{k-1} \beta_j A_0 \varphi_{j-k+1} + \sum_{j=0}^{k} \beta_j A_1 \varphi_{j-k-N+1} \right). \tag{2.26}$$

Finally, we solve for ψ_0 by rearranging the terms,

$$\psi_0 = R^{-1} \left(\sum_{j=0}^{k-1} (-\alpha_j I + h\beta_j A_0) \varphi_{j-k+1} + \sum_{j=0}^{k} h\beta_j A_1 \varphi_{j-k-N+1} \right), \tag{2.27}$$

where we let $R = I - h\beta_k A_0$ for notational convenience.

Note that the last sum contains terms which are outside of the interval of φ. This method assumes that the shift property holds for these as well.

Similar to the example, we may now combine the shift operation and the approximation of the differential part and get a matrix as an approximation of the solution operator

$$\begin{pmatrix} \psi_{-N-k+1} \\ \psi_{-N-k+2} \\ \vdots \\ \psi_0 \end{pmatrix} = \begin{pmatrix} 0 & I & & \\ & 0 & I & \\ & & \ddots & \ddots \\ & & A^T & \end{pmatrix} \begin{pmatrix} \varphi_{-N-k+1} \\ \varphi_{-N-k+2} \\ \vdots \\ \varphi_0 \end{pmatrix},$$

where the last row and the coefficients $A^T \in \mathbb{R}^{n\times(n(N+k))}$ are given by (2.27). More precisely,

$$A^T = (A_0^T, 0_{n\times n(N-k-1)}, A_1^T),$$

where $0_{j\times k} \in \mathbb{R}^{j\times k}$ is the zero matrix,

$$A_0^T = h(\beta_0, \ldots, \beta_k) \otimes (R^{-1}A_1) \in \mathbb{R}^{n\times n(k+1)},$$

and

$$A_1^T = -(\alpha_0, \ldots, \alpha_{k-1}) \otimes R^{-1} + h(\beta_0, \ldots, \beta_{k-1}) \otimes (R^{-1}A_1) \in \mathbb{R}^{n\times nk}.$$

Now consider the eigenvalue problem $S_N x = \mu x$. This is a companion linearization of a polynomial eigenvalue problem. Companion linearizations are described in Section A.1. The matrix coefficients of the polynomial eigenvalue problem are given by the matrices in A^T. A simple form can be found by considering (2.26) directly. Note that $\psi_p = \mu^{p+N+k}u$, $p = -N-k, \ldots, 0$, if u is the eigenvector of the polynomial eigenvalue problem. It now follows from (2.26) and the definition of f (2.23) that the polynomial eigenvalue problem is

$$\sum_{j=0}^{k} \alpha_j \mu^{j+N} u = h \sum_{j=0}^{k} \beta_j \left(A_0 \mu^{j+N} + A_1 \mu^j\right) u. \tag{2.28}$$

In the alternative derivation presented below we will use the order of the LMS-method, which is defined as follows. In our context, it turns out to be advantageous to define the order using the shift operator E, i.e., $(Ef)(x) = f(x+h)$ (and $(E^\alpha f)(x) = f(x+\alpha h)$) and the differentiation operator D, i.e., $Df = f'$.

Definition 2.14 *The linear multistep method defined by the two characteristic polynomials*[3] $\alpha(s) = \alpha_0 + \cdots + \alpha_k s^k$, $\beta(s) = \beta_0 + \cdots + \beta_k s^k$ *is of order* p *if*

$$\alpha(E)z - \beta(E)hDz = C_{p+1}(hD)^{p+1}z + C_{p+2}(hD)^{p+2}z + \cdots,$$

for some constants C_i, $i = p+1, p+2, \ldots$, *and any function* $z \in \mathcal{C}^1[a,b]$.

[3] In the literature, these polynomials are often referred to as the *first* and *second* characteristic polynomials and are denoted $\rho(s)$ and $\sigma(s)$.

SOD(LMS) is a Padé approximation of the logarithm

The solution operator is a characterization of the solution of the DDE. We are interested in the eigenvalues and not the solution of the DDE. Hence, the derivation using the solution operator appears somewhat indirect. We now wish to search alternative shorter derivations leading to (2.28) without using the solution operator. The value of this discussion that follows is the simplicity of the derivation and that we point out that the resulting approximation is a polynomial eigenvalue problem. Polynomial eigenvalue problems can be solved using other methods than the companion linearization which was implied by the discretization motivation above. There are several methods for polynomial eigenvalue problems, e.g. the different types of linearizations [MMMM06b] and Jacobi-Davidson for polynomial eigenvalue problems [SBFvdV96] (see also Section 2.3.2).

It turns out that the linear multistep discretization of the solution operator (SOD(LMS)) is equivalent to a Padé-approximation of the logarithm. The relation between ODE-methods and polynomial (and rational) interpolation is not new (see e.g. [Lam91, Section 3.3]). It is however worthwhile to clarify the interpretation in the current context, i.e., how LMS applied to the solution operator can be interpreted as a rational approximation.

The substitution $s = \frac{1}{h}\ln(\mu)$ for $h = 1/N$ into the characteristic equation yields

$$\frac{1}{h}\mu^N \ln\mu \in \sigma(A_0\mu^N + A_1). \tag{2.29}$$

We are interested in $\mu \approx 1$, as this corresponds to s close to the origin. We will now approximate the logarithm close to this point. For instance, consider the first-order Taylor expansion $\ln\mu \approx \mu - 1$. This yields

$$\frac{1}{h}\tilde{\mu}^N(\tilde{\mu} - 1) \in \sigma(A_0\tilde{\mu}^N + A_1),$$

which reduces to Example 2.12 for $A_0 = 0$, $A_1 = -3/2$.

Similarly, we may consider a rational approximation of $\ln\mu$, i.e.,

$$\ln\mu \approx \frac{\alpha(\mu)}{\beta(\mu)},$$

where α, β are polynomials (where we denote the polynomials α and β as they will coincide with the polynomials in Definition 2.14). If we insert this approximation in (2.29) we have derived (2.28).

For instance, the $(2,2)$-Padé approximation of $\ln(\mu)$ in $\mu = 1$ is

$$\ln \mu \approx \frac{\mu^2 - 1}{\frac{1}{3}\mu^2 + \frac{4}{3}\mu + \frac{1}{3}}. \tag{2.30}$$

See for instance [Hig01] for Padé approximations of $\ln(1+x)$. The default LMS-method in DDE-BIFTOOL 2.03 is Milne-Simpson. It is not a coincidence that the characteristic polynomials for the LMS-method Milne-Simpson, given by $\alpha(s) = s^2 - 1$ and $\beta(s) = (s^2 + 4s + 1)/3$, are the numerator and denominator of the approximation (2.30).

In fact, an LMS-method of order p with characteristic polynomials α, β is a pth order Padé approximation α/β of the logarithm at $\mu = 1$. We have the following motivation. The shift-operator E and the differentiation operator D are (formally) related by $hD = \ln(E)$ (see [Bic48] or [DB08, Chapter 3]).

Multiplying the order definition (Definition 2.14) by $\beta(E)^{-1}$, yields

$$\beta(E)^{-1}\alpha(E)z - \ln(E)z = C_{p+1}\beta(E)^{-1}(hD)^{p+1}z + C_{p+2}\beta(E)^{-1}\frac{(hD)^{p+2}}{z} + \cdots . \tag{2.31}$$

If α, β correspond to an LMS-method of order p, (2.31) holds for any function z, in particular for $z(x) = e^x$. Suppose $\mu = e^h$, we conclude that

$$\frac{\alpha(\mu)}{\beta(\mu)} - \ln(\mu) = \mathcal{O}(h^{p+1}) = \mathcal{O}((1-\mu)^{p+1}),$$

which is the Padé approximation α/β of order p in point $\mu = 1$.

LMS(RK) is a also a polynomial eigenproblem

The ODE-segment of the solution operator was approximated with a linear multistep discretization in SOD(LMS). General Runge-Kutta discretizations can also be used. This approach is taken by D. Breda in [Bre06]. Without attempting a complete characterization, we now wish to point out that the method in [Bre06] also results in a polynomial eigenvalue problem.

Breda considers the class of s-stage Runge-Kutta schemes (A, b, c) where $A \in \mathbb{R}^{s\times s}$, $b^T = (a_{s1}, a_{s2}, \cdots, a_{ss})$ and $0 < c_1, \cdots < c_s = 1$. Unlike the LMS-method

each interval $[\theta_j, \theta_{j+1}]$ is also discretized. The discretization $S_N \in \mathbb{R}^{nsN \times nsN}$ of the solution operator is given by

$$S_N = \begin{pmatrix} 0 & I & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & 0 & I \\ hR(hA_0)(A \otimes A_1) & 0 & \cdots & 0 & R(hA_0)(1_s e_s^T \otimes I_m) \end{pmatrix},$$

where

$$R(Z) = (I - A \otimes Z)^{-1} \text{ and } 1_s = (1, \cdots, 1)^T.$$

This is also the companion matrix of the polynomial eigenvalue problem of dimension $ns \times ns$,

$$(hR(hA_0)(A \otimes A_1) + R(hA_0)(1_s e_s^T \otimes I_m)\mu^{N-1} - \mu^N I)u = 0,$$

or

$$\left(h(A \otimes A_1) + (1_s e_s^T \otimes I_m)\mu^{N-1} - (I - A \otimes hA_0)\mu^N\right) u = 0. \tag{2.32}$$

It is worthwhile to determine in what sense (2.32) is an approximation of (2.29). Equation (2.32) can be derived from (2.29) by multiplying (with the Kronecker product) from the left with $A \in \mathbb{R}^{s \times s}$ and approximating

$$A \ln(\mu) \approx I - 1_s e_s^T \mu^{-1}. \tag{2.33}$$

The approximation is of course only the formal relation between (2.29) and (2.32) and the approximation error of (2.33) is (except for $s = 1$) large. It is somewhat remarkable that this approximation yields a successful efficient method despite of the fact that the approximation (2.33) appears crude. It is shown in [Bre06] that the convergence order is p/ν, where p is the order of the Runge-Kutta scheme and ν is the multiplicity of the eigenvalue. More precisely, [Bre06, Theorem 4] essentially states that, if $s_* \in \sigma(\Sigma)$ is an eigenvalue with multiplicity ν then there is a step-length h such that $\det(-sI + S_N) = 0$ has ν roots, $s_1, \ldots, s_\nu$ for which

$$\max_{i=1,\ldots,\nu} |s_* - s_i| = \mathcal{O}(h^{p/\nu}).$$

Breda also addresses multiple delays by Lagrange interpolation and a Gauss-Legendre quadrature formula for distributed delays. The order of the interpolation and the quadrature formulas are chosen high enough to ensure that the accuracy order of the combined method is dominated by the discretization and not the interpolation and quadrature approximations.

Chebyshev discretization of the solution operator

Finally, we will briefly discuss yet another method to discretize the solution operator. This method is presented in series works of Butcher, Bueler, et al. [BMB+04], [Bue04] and [Bue07] and implemented in the recent Matlab-package `ddec` by the same authors, where the authors consider a generalization of DDEs, DDEs with periodic coefficients. Independently, an equivalent method for DDEs with constant coefficients was published and analyzed in the thesis of D. Breda [Bre04, Section 3.3.3]. This was later extended to DDEs with periodic coefficients and presented in the conference proceedings [BMV06a]. To the author's knowledge, the first work of Breda and that of Butcher, Bueler et al., were parallel independent results.

It is illustrative to start by presenting a couple of lines of Matlab-code for our purpose, i.e., the single delay DDE.

We construct the discretization by using routine `cheb` to compute the (so-called) Chebyshev differentiation matrix from [Tre00][4]. The following lines of code (following the description of matrices in [Bue07]) compute some eigenvalues of the single DDE of dimension n with N Chebyshev nodes.

```
DD=cheb(N-1)*2/tau;
DN=kron([DD(1:end-1,:);[zeros(1,N-1),1]],eye(n));
MA=kron([eye(N-1,N);zeros(1,N)],A0);                    (2.34)
MB=[kron([eye(N-1,N)],A1);kron([1,zeros(1,N-1)],eye(n))];
(log(eig(MB,DN-MA))+k*2*pi*i)/tau
```

Note that we must choose the correct branch k of the logarithm in the last step. This stems from the fact that the substitution $\mu = e^{-\tau s}$ is not bijective, e.g. $e^{-\tau s} = e^{-\tau(s+2\pi i/\tau)}$. The rightmost eigenvalues are typically the principal branch $k = 0$. Note that the real part of the eigenvalues are (fortunately) independent of k. Hence, the choice of branch is not relevant in a stability analysis.

The setting in the works by Bueler, Butcher, et al. as well as [BMV06a] is much more general than in the code stated above. The authors consider DDEs with periodic coefficients (see [HV93, Chapter 8]). Constant coefficients (as we consider in this work) are trivially periodic. In [Bue07] and [BMV06a] the method

[4] `cheb.m` is also available online:
http://www.comlab.ox.ac.uk/nick.trefethen/spectral.html

is used to succesfully compute the stability chart of the *delayed Mathieu equation*

$$\ddot{x}(t) + (\delta + \varepsilon \cos(t))x(t) = bx(t - 2\pi).$$

The stability chart of the delayed Mathieu equation is also the topic of [IS02], where an Fourier expansion of the periodic solution to find the boundary curves for a fixed ε. In [Bue07] the author presents an error analysis by constructing a method to compute *a posteriori* error bounds of a Chebyshev spectral collocation method using a generalization of the Bauer-Fike theorem to Hilbert spaces. As expected, the convergence of this method is exponential. More precisely, [Bue07, Theorem I] states that if $s \in \sigma(\Sigma)$ and s_i are the approximations, then

$$\min_i |s - s_i| \leq \omega \operatorname{cond}(\tilde{V}),$$

where $\operatorname{cond}(\tilde{V})$ is the condition number of the approximation of the discretization of the solution operator. Bueler observes exponential convergence decay in ω and a slow increase in the condition number as the number of discretization points N is increased. Exponential convergence is proven for several examples.

The terminology in the field of DDEs and particular periodic coefficient DDEs is unfortunately ambiguous. This is particularly clear in the works we just mentioned. Note that the *(Floquet) multipliers* of the *monodromy operator* (in the terminology of e.g. Bueler and [HV93, Chapter 8]) for DDEs with constant coefficients, are the eigenvalues μ of the solution operator, in the terminology we mostly use.

Example 2.15 (Chebyshev discretization of the solution operator) *We now apply* (2.34) *to Example 2.12 for some number discretization points N. In the table below one of the left-most eigenvalues is computed with the Chebyshev discretization. The correct digits are marked with bold. The error is $1 \cdot 10^{-15} \approx 10\epsilon_{mach}$, for $N = 13$. We give more comments on these results in Example 2.19.*

$N = 3$	$N = 5$	$N = 13$	$W_0(-3/2)$ (exact)
$\mathbf{-0.0323} + \mathbf{1}.6354i$	$\mathbf{-0.0327 + 1.5493}i$	$\mathbf{-0.0328 + 1.5496}i$	$\mathbf{-0.0328 + 1.5496}i$

2.2.3 PDE discretization (IGD)

Every DDE can be rewritten as a *partial differential equation* (PDE) by introducing an additional dimension, containing the function segment x_t to the left

(in time) of the time-point t. This introduced *memory-dimension* θ takes the position of the space-dimension in the PDE-formulation. The PDE-formulation is extensively used in [HV93] and [DvGVW95].

Another set of methods to compute the eigenvalues are based on discretizing the PDE. We will refer to this type approach as the *infinitesimal generator discretization* (IGD) as the operator associated with the PDE-formulation is in fact the infinitesimal generator of the solution operator.

PDE formulation of a DDE

We now present the PDE-formulation and show how it is related to the infinitesimal generator of (the semigroup corresponding to) the solution operator. The single-delay DDE

$$\begin{cases} \dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) & t \geq 0 \\ x(t) = \varphi(t) & t \in [-\tau, 0] \end{cases} \tag{2.35}$$

can be rewritten as the boundary value problem (BVP)

$$\begin{array}{lll} \text{PDE:} & \frac{\partial u}{\partial \theta} = \frac{\partial u}{\partial t} & t \geq 0,\ \theta \in [-\tau, 0], \\ \text{BV:} & u'_\theta(t,0) = A_1 u(t,-\tau) + A_0 u(t,0) & t \geq 0, \\ \text{IC:} & u(0,\theta) = \varphi(\theta) & \theta \in [-\tau, 0] \end{array} \tag{2.36}$$

for the unknown $u \in \mathcal{C}([0,\infty) \times [-\tau, 0], \mathbb{R}^n)$. The PDE formulation of the DDE is equivalent to the DDE in the following sense.

Theorem 2.16 (PDE-formulation) *Let $\varphi \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$ be given. Suppose $x(t)$ is the solution to (2.35) and $u(t,\theta)$ a solution to (2.36), then*

$$u(t,\theta) = x(t+\theta), \tag{2.37}$$

for $\theta \in [-\tau, 0]$, $t \geq 0$.

Proof: We show both directions by first using a solution $x(t)$ of (2.35) to define u using (2.37) and showing that u then fulfills (2.36). The converse is shown by using a solution $u(t,\theta)$ of (2.36), defining $x(t)$ by (2.37) and proving that x the fulfills (2.35).

Let $x(t)$ be a solution to (2.35) and $u(t,\theta) := x(t+\theta)$. We now have that

$$\frac{\partial u}{\partial \theta} = \frac{\partial u}{\partial t}$$

since $x(t+\theta)$ is symmetric with respect to t and θ. It remains to show that the boundary condition holds. Note that $u'_\theta(t,0) = \dot{x}(t)$, and hence $u'_\theta(t,0) = A_0x(t) + A_1x(t-\tau) = A_0u(t,0) + A_1u(t,-\tau)$.

The converse is proven next. Suppose $u(t,\theta)$ is a solution to (2.36) and let $x(t) := u(t,0)$ for $t \geq 0$. From (2.36) it follows that $\dot{x}(t) = u'_t(t,0) = u'_\theta(t,0) = A_0u(t,0) + A_1u(t,-\tau) = A_0x(t) + A_1x(t-\tau)$. □

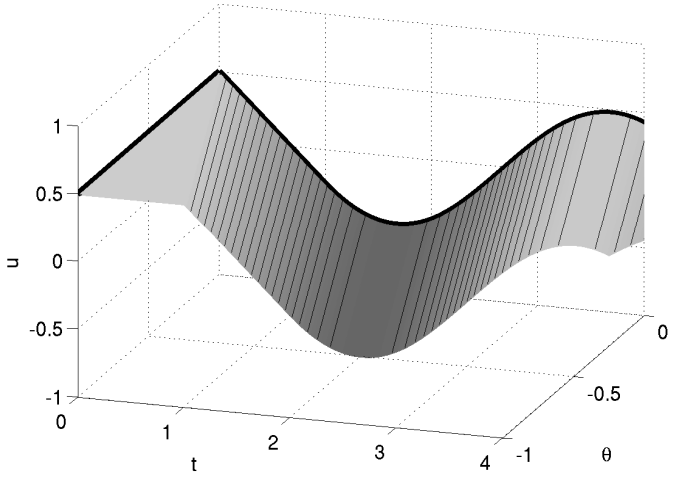


Figure 2.4: The boundary value problem for the DDE in Example 2.17. The thick lines are initial conditions $\varphi(\theta)$ and solution $x(t)$.

The boundary value problem (2.36) is a transport equation with interconnected (nonlocal) boundary conditions. That is, the boundary conditions are non-local in the sense that there is only one boundary condition, but it is expressed in terms of both sides, $\theta = -\tau$ and $\theta = 0$ and the derivative at $\theta = 0$. The same type of formulation carries over to multiple delays as well as distributed delays. DDEs with distributed delays contain an integral term of $x(t-\tau)$ over

a delay-interval. In case of multiple or distributed delays, the "boundary conditions" do not only depend on the boundary, but on interior points as well. We will restrict the discussion to single delays for simplicity.

Example 2.17 (PDE-formulation) *Consider the DDE*

$$\begin{cases} \dot{x}(t) = -\frac{3}{2}x(t-1) & t \geq 0 \\ x(t) = \varphi(t) = 0.5 & t \in [-1, 0]. \end{cases} \tag{2.38}$$

The PDE-representation of this DDE is visualized in Figure 2.4. The boundaries are the shifted solutions $x(t)$, cf. Figure 2.2.

The PDE is a linear infinite-dimensional system with $x_t(\cdot)$ as the state. Let $\mathcal{A}$ correspond to the differentiation operator in θ-direction with the domain of functions fulfilling the boundary conditions in (2.36) (sometimes referred to as the *splicing condition*). That is

$$(\mathcal{A}\varphi)(\theta) := \frac{d\varphi}{d\theta}(\theta), \tag{2.39}$$

for a function φ which fulfills $\varphi'(0) = A_1\varphi(-\tau) + A_0\varphi(0)$. The boundary value problem (2.36) is hence,

$$\frac{d}{dt}x_t = \mathcal{A}x_t, \tag{2.40}$$

which is sometimes referred to as an *abstract Cauchy-problem*. We have here defined $\mathcal{A}$ as the infinite-dimensional operator corresponding to the BVP (2.36). In the literature (e.g. in standard references such as the book by Hale and Verduyn Lunel [HV93]), the definition is often presented using the solution operator T. More precisely,

$$\mathcal{A}\varphi := \lim_{t \to 0^+} \frac{1}{t}(T(t)\varphi - \varphi).$$

This is the definition of the *infinitesimal generator* for a $\mathcal{C}^0$-semigroup (or *strongly continuous semi-group*) corresponding to T. The set of operators $\{T(t)\}_{t \geq 0}$ is indeed a $\mathcal{C}^0$-semigroup as the solution operator T fulfills $T(0) = I$, $T(t+s) = T(t)T(s)$ and for any φ, $\lim_{t \to 0^+} \|T(t)\varphi - \varphi\| \to 0$. Moreover, the infinitesimal generator of T is indeed the differentiation operator in θ-direction (see e.g. [HV93, Chapter 7,Lemma 1.2]).

Discretizing the PDE

The eigenvalues of the operator $\mathcal{A}$ are the eigenvalues of the DDE (see e.g. [HV93, Chapter 7, Lemma 2.1]). The idea is now to discretize $\mathcal{A}$, i.e., the PDE, in θ-direction (space) and compute the eigenvalues of the corresponding finite-dimensional linear operator (matrix) A_N. A number of discretizations of (2.40) have been addressed in the literature, e.g. a forward difference[5] [BM00], Runge-Kutta discretizations [BMV04] (see also [Bre04]), linear multistep [Bre04] and spectral differencing [BMV05b] and [BMV06b]. We will now derive some of the methods, and give new interpretations of some of them.

The first (crude) approximation we can think of is to replace the differential (2.39) by a finite difference, e.g. forward difference. This discretization was used on (2.39) in [BM00] (see also [BZ03, Section 3.5]), but with the purpose to construct a method to solve the DDE.

Let $u_N(t) \in \mathbb{R}^n$ be the vector approximating x_t in some discretization points, i.e., $u(t, \theta_j) = (u_N(t))_j$ for the discretization points θ_j. If we approximate the derivative in θ with a forward difference on a uniform grid, the abstract Cauchy-problem (2.40) turns into a linear system of ODEs

$$\dot{u}_N(t) = A_N u_N(t), \quad u_N(0) = \varphi_N,$$

where

$$A_N = \begin{pmatrix} D_N \otimes I_n \\ A_1 \; 0 \cdots 0 \; A_0 \end{pmatrix}, \tag{2.41}$$

and

$$D_N = \frac{1}{h} \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{N \times (N+1)}.$$

Note that the first block row of A_N is size $nN \times (n(N+1))$ corresponding to the differentiation, and the last block row (of size $n \times n(N+1)$) corresponds to the boundary condition. We will refer to this method as IGD(Euler) as it corresponds to the *Euler method* (for ODEs) applied to the PDE (2.36).

[5] The main results are based on forward difference discretization but the authors point out that the same procedure can be performed for higher order methods.

Example 2.18 (Forward discretization of the PDE, IGD(Euler)) *We again consider the scalar example, Example 2.17. We approximate the eigenvalues of $\mathcal{A}$, i.e., the eigenvalues of the DDE by the eigenvalues of A_N (2.41) for some choices of N. The error of the approximation of the rightmost eigenvalue for different N are shown below. The approximation is so crude that the error is fairly large ($2 \cdot 10^{-3}$) even for $N = 500$. This method is clearly more computation demanding than Example 2.15 where $N = 13$ was sufficient to get (almost) machine precision.*

$N = 10$	$N = 100$	$N = 500$	$W_0(-3/2)$ (exact)
$-\mathbf{0}.1095 + \mathbf{1}.4923i$	$-\mathbf{0.0}410 + \mathbf{1.54}37i$	$-\mathbf{0.03}44 + \mathbf{1.54}85i$	$\mathbf{-0.0328 + 1.5496i}$

We now wish to show how this method can be interpreted as a direct approximation applied to the characteristic equation (2.2). As mentioned in the introduction, it turns out that if we truncate the limit in the definition of the exponential we get the approximation (2.41). That is, we replace $e^{-\tau s} = \lim_{N\to\infty}(1+\tau s/N)^{-N}$ with $e^{-\tau s} \approx S_N(s) := (1+\tau s/N)^{-N}$. We can derive (2.41) as follows. Let $h = \tau/N$. If we replace $e^{-\tau s}$ with $(1+hs)^{-N}$ and multiply the characteristic equation (2.2) by $(1+hs)^N$ we have

$$(A_0(1+hs)^N + A_1)v = s(1+hs)^N v. \tag{2.42}$$

We now define the vectors $v_k = (1+hs)v_{k-1}$ recursively with $v_1 = v$. Clearly $v_N = (1+hs)^N v$. The constructed vector $w = (v_1^T, \dots v_N^T)^T$ is now an eigenvector of (2.41). The first block row is fulfilled from the recursive construction of v_k and the last row is (2.42). In the context of linearizations of polynomial eigenvalue problems, (2.41) is a (somewhat unusual) linearization of (2.42). See [MMMM06b] for a characterization of linearizations.

The forward difference is of course a very crude approximation and is in practice (as we will see in the examples) not very useful. But the general idea remains unchanged for other discretizations. That is, the same procedure can be used for other PDE discretization methods as well, e.g. Runge-Kutta [BMV04] and pseudospectral[6] differencing techniques [BMV05b].

In principle D_N in (2.41) can be replaced with any differentiation matrix. A

[6] Pseudospectral methods are not really related to pseudospectra. Pseudospectral methods are spectral collocation methods (see [Tre00]).

particularly popular and successful approximation is the *Chebyshev differentiation matrix*.

We wish to stress the simplicity to implement this method (i.e., [BMV05b]) with modern software. We will, similar to the SOD-method (2.34) use Matlab and the routine `cheb` to compute the Chebyshev differentiation matrix from [Tre00]. The method can be elegantly implemented in two lines of code:

```
D=-cheb(N-1)*2/tau;
eig([kron(D(1:N-1,:),eye(n));[A1,zeros(n,(N-2)*n), A0]]);
```
(2.43)

The first line determines the Chebyshev differentiation matrix for the interval $[-\tau, 0]$. We scale with $-2/\tau$ because `cheb` returns the differentiation matrix for the interval $[-1, 1]$ with descending ordering of the interpolation points. We switch to ascending ordering by multiplying with -1, in order to blend with (2.41). We will refer to the this class of methods as IGD(PS), *pseudospectral discretization* of the infinitesimal generator.

The boundary conditions are imposed by ignoring the last row in the differentiation matrix. There are other ways to impose the boundary conditions which will not be necessary for our purposes, e.g. one may add columns to the differentiation matrix, cf. [Tre00].

Note that the construction of the discretization of the solution operator in (2.34) (Bueler et al. [Bue07]) does not seem to be equivalent to the construction of Breda, i.e., (2.43). For the example below the computational effort for the two Chebyshev methods are however very similar.

Example 2.19 (Chebyshev differentiation matrix IGD(PS)) *We again consider the scalar problem in Example 2.17, i.e., $A_0 = 0$, $A_1 = -3/2$, $\tau = 1$. The table below shows the approximation of one eigenvalue using the method with the Chebyshev differentiation matrix described above, i.e., the two lines of code* (2.43)*. For $N = 5$ the error is $6 \cdot 10^{-4}$ which is sufficient for many applications. For $N = 13$ we have reached 10^{-15} which is $10\epsilon_{mach}$. Clearly, IGD(PS) is here much more efficient than IGD(Euler) (Example 2.18) where we computed the rightmost eigenvalues to an accuracy of $2 \cdot 10^{-3}$ by computing the eigenvalues of a matrix of dimension 500 whereas we here reached (essentially) machine-precision from a matrix of dimension 13. It is interesting to note that the Chebyshev discretization method in Example 2.15 we also needed exactly $N = 13$ to achieve an accuracy of 10 times machine precision.*

$N = 3$	$N = 5$	$N = 13$	$W_0(-3/2)$ (exact)
$\mathbf{-0.0}702 + \mathbf{1}.4468i$	$\mathbf{-0.032}6 + \mathbf{1.5}502i$	$\mathbf{-0.0328 + 1.5496i}$	$\mathbf{-0.0328 + 1.5496i}$

Other approximations of the exponential

We saw above that the discretization of the infinitesimal generator could be interpreted as an approximation of the exponential function, at least for some cases. In principle, one could take any method from rational (analytic) interpolation or approximation to construct a method to approximate the eigenvalues of a DDE. In the context of DDEs, rational approximations have been used to approximate the exponential in a number of articles by Partington, Mäkilä and Bonnet [Par91], [MP99a], [MP99b], [Par04c], [PB04], [PM05], and [BP07]. See also the monograph [Par04a, Chapter 6]. Even though most of the results are not directly methods for the spectrum of the DDE, many results carry over naturally to our context. The results are mostly related to approximations of the transfer function of a dynamical system with delay. We describe the general idea with the example in [MP99b]. The authors approximate $G(s) = e^{-hs}/(Ts+1)^5$, where T is a model parameter. The approximation is based on the Padé-approximation with shift

$$e^{-hs} \approx u_n(s) := \frac{(1 - hs/(2n) + h^2s^2/12n^2)^n}{(1 + hs/(2n) + h^2s^2/12n^2)^n}. \tag{2.44}$$

They present a bound of the error (in H_∞-norm) which is verified with the numerical example. The bound in [MP99b] is for functions of the form $e^{-hs}R(s)$ where R is a rational function. An explicit bound for the exponential approximation is given in [PM05]. It is shown that

$$\left| e^{-z} - \frac{p(-z)}{p(z)} \right| \leq \frac{1}{270}|z|^5, \ |z| \leq 1,$$

where $p(z) = 1+z/2+s^2/12$, such that $p(-z)/p(z)$ is the approximation in (2.44) for $h = 1$ and $n = 1$. Now the approximation error of $u_n(z) = (p(-z/n)/p(z/n))^n$ is bounded by

$$|e^{-z} - u_n(z)| \leq \frac{1}{270} \left| \frac{z}{n} \right|^5 n3^{-\mathrm{Re}\ z}, \text{ if } |z/n| \leq 1 \text{ and } n > -\mathrm{Re}\ z.$$

We have seen that time-delay systems can be treated as infinite dimensional system. The rational approximation is hence a *model order reduction* from infinite

order to the order of the rational function. Several rational approximation techniques, including Fourier-Laguerre series, shift-based approximations, Malmquist bases and wavelet-based techniques, in the context of model order reduction are reviewed in [Par04c]. See also [Rei05] for some notes on a different model order reduction approach, where the approximation is similar to that of Example 2.18, i.e., here IGD(Euler). Note that several issues related to model reduction for time-delay systems seem to be open problems [Par04b].

2.3 Methods for nonlinear eigenvalue problems

The problem of finding the values $s \in \mathbb{C}$ such that a parameter dependent matrix $T(s)$ is singular is sometimes called a *nonlinear eigenvalue problem*. Equivalently, we search for $s \in \mathbb{C}$ and $v \in \mathbb{C}^n \backslash \{0\}$ such that

$$T(s)v = 0. \tag{2.45}$$

We call s an eigenvalue and v the right eigenvector (or sometimes just eigenvector) of the nonlinear eigenvalue problem. The nonlinear eigenvalue problem (2.45) is indeed a numerically and computationally challenging problem ([MV04]). It is a generalization of the root-finding $f(s) = 0$ as well as the standard eigenvalue problem $sv = Av$. Root-finding problems are typically solved with fixed point iterations such as Newton-iteration, whereas the eigenvalue problem is typically solved with variants of the QR-method, inverse iteration or projection methods such as Arnoldi or Jacobi-Davidson. It is not surprising that all the methods mentioned above have been used in the literature to derive methods for the nonlinear eigenvalue problem (2.45).

In general, methods motivated by the corresponding root-finding method (e.g. *MSLP* and Newton-iteration below) have good local convergence properties. That is, given a good starting guess, these methods typically converge in a reliable way with reasonable computational cost. These types of methods are discussed in Section 2.3.1.

Clearly, local convergence properties are not sufficient for many problems. For instance, in a stability analysis, the rightmost eigenvalues of (2.45) should be computed. In this context, missing an eigenvalue is not acceptable. Another typical problem is to find the eigenvalues closest to some target $\sigma_t \in \mathbb{C}$. For some nonlinear eigenvalue problems of small or moderate dimension, there

are methods with good global convergence properties. For instance, polynomial eigenvalue problems can be solved by linearization (see below), the delay eigenvalue problem can be solved by the methods in Section 2.2. The general idea in the projection type methods, which we discuss in Section 2.3.2, is to inherit the good global convergence properties of a small nonlinear eigenvalue problem constructed from the projection of a large problem onto a search-space. We wish to stress that methods of this type have proven extremely efficient and have been used to solved previously unsolved problems, but typically unfortunately contain heuristic ingredients which are not (yet) completely mathematically understood.

The case that T is a matrix polynomial[7] (of degree m) is particularly interesting, as it is possible to transform the problem to a generalized eigenvalue problem (of dimension $n(m-1)$) by so-called *linearization*, cf. [MMMM06b], where the companion linearization is the most popular method. There are also methods which exploit that T is a matrix polynomial without linearizing it, e.g. the generalization of the Arnoldi method [BS05].

In this section we will not further discuss the matrix polynomial, but focus on general methods without being exhaustive. See [Ruh73] and more recently [MV04] for more complete lists of methods.

2.3.1 Scalar and vector valued fixed point methods

Kublanovskaya's QR-method

The nonlinear eigenvalue problem is mathematically equivalent to finding the roots of the characteristic equation, i.e.,

$$g(s) = \det(T(s)) = 0.$$

A straightforward approach to this problem is to apply a root-finding method (say Newton) to g. We know from the theory of eigenvalue problems that such methods are likely to have a very small domain of attraction and are likely to be very sensitive to rounding errors and computationally demanding, at least for large problems. Some of these problems can be circumvented by considering a slightly different definition of the target function g containing the QR-decomposition of T, i.e., the decomposition of $T(s)$ into two (parameter dependent) matrices $Q(s)$ and $R(s)$ where $Q(s)$ is orthogonal and $R(s)$ an upper triangular matrix.

[7] A matrix polynomial is sometimes called a *lambda-matrix*.

Suppose $Q(s)R(s) = T(s)P(s)$ is the QR-decomposition of $T(s)$ where $P(s)$ is permutation matrix such that $r_{nn}(s)$ is the smallest element in magnitude of the diagonal elements of $R(s)$. Then,

$$g(s) = \pm \det(R(s)).$$

Clearly, $g(s) = 0$ if and only if $r_{nn}(s) = 0$. The idea in the method of Kublanovskaya [Kub70] is to apply Newton's method to $f(s) = r_{nn}(s)$, i.e.,

$$s_{j+1} = s_j - f(s_j)/f'(s_j).$$

Kublanovskaya showed that the inverse of the correction quotient is given by

$$f'(s)/f(s) = e_n^T Q(s)^H T'(s) P(s) R(s)^{-1} e_n,$$

which allows the possibility to efficiently compute the correction term. Unfortunately, the eigenvalue paths of a parameter dependent matrix are continuous but in general not differentiable. The elements of R are not in general differentiable. Hence, Kublanovskaya's Newton iteration does not have quadratic convergence in general. This as well as ambiguities in Kublanovskayas derivation were pointed out together with a corrected method in [JS83].

There are similar results for the LU-decomposition of $T(s)$. This approach is taken in [JSH83], yielding a method which also allows the computation of eigenvalue sensitivities with hardly any extra computational effort.

Method of successive linear problems (MSLP)

Suppose $s_* \in \mathbb{C}$ is a solution to the nonlinear eigenvalue problem (2.45). Then the Taylor expansion of T around some point $s_k \in \mathbb{C}$ evaluated at s_* is

$$0 = T(s_*)v = T(s_k)v + (s_* - s_k)T'(s_k)v + \mathcal{O}(s_* - s_k)^2.$$

If we neglect the higher order terms we (hopefully) get a better approximation of $s_* \approx s_{k+1}$ from the generalized eigenvalue problem,

$$s_{k+1} T'(s_k) v^{(k)} = (s_k T'(s_k) - T(s_k)) v^{(k)}. \tag{2.46}$$

That is, the next iterate s_{k+1} can be computed from a generalized eigenvalue problem (2.46) where the matrices depend on s_k. This iteration is referred to

as the *method of successive linear problems* (MSLP) [Ruh73]. While deriving (2.46) we neglected the quadratic term, it is hence not surprising that the convergence is locally quadratic to simple eigenvalues [Vos04b]. Convergence order of such iterations is the topic of Chapter 4. In particular, we verify the quadratic convergence of MSLP with Theorem 4.10.

Certainly, this is not the only way to construct a sequence of linear problems which locally converges to the solution of the nonlinear eigenvalue problem. For instance, consider the continuous eigenvalue paths $\lambda_i(s)$ of $T(s)$, i.e., $\cup_i \lambda_i(s) = \sigma(T(s))$. Clearly, $\lambda_i(s_*) = 0$ for some i as $T(s_*)$ is singular. We can now apply a root-finding method to λ_i. Suppose $v^{(k)}$, $u^{(k)} \in \mathbb{C}^n$ are the right and left eigenvectors corresponding to the eigenvalue $\lambda_i(s_k)$ of $T(s_k)$. Then the Newton iteration applied to $\lambda_i(s)$ is

$$s_{k+1} = s_k - \frac{\lambda_i(s_k)}{\lambda_i'(s_k)} = s_k - \frac{u^{(k)*}T(s_k)v^{(k)}}{u^{(k)*}T'(s_k)v^{(k)}} \tag{2.47}$$

as $\lambda_i'(s) = u^{(k)*}T'(s)^{(k)}v$, with the normalization $u^{(k)*}v^{(k)} = 1$, cf. [Lan02, Theorem 2.5]. This approach is named *Newton-Raphson* [Lan02, Section 5.4]. It is illustrative to rearrange the terms of (2.47) to compare it with (2.46). If we multiply (2.47) by $v^{(k)}u^{(k)*}T'(s_k)v^{(k)}$, then

$$s_{k+1}v^{(k)}u^{(k)*}T'(s_k)v^{(k)} = v^{(k)}u^{(k)*}\left(s_kT'(s_k) - T(s_k)\right)v^{(k)}.$$

The method proposed by Lancaster is hence a variant of MSLP where only a rank one part is taken into account. An essential difference between MSLP and the method of Lancaster is that in one step of MSLP a generalized eigenvalue problem must be solved (2.46) whereas in Lancaster's method the eigenvalues of the matrix $T(s_k)$ must be computed. Since it is slightly more computationally expensive to solve generalized eigenvalue problems than computing the eigenvalues of a matrix, one step of MSLP is slightly more expensive than (2.47). Both (2.46) and (2.47) have quadratic convergence. However, which one is better in terms of convergence domain and rate, is an open problem.

A method of Yang [Yan83] is also based on a Taylor expansion and successive linear problems. Yang suggests to determine the smallest eigenvalue, which corresponds to the smallest correction, by using the elements of an LU-decomposition to construct an approximate solution of the linear eigenvalue problem. This was applied to numerically analyze problems in vibration and structural stability systems in [SR02].

An approach based on a sequence of linear problems was also taken in [Thu78] where the update $\Delta = s_{k+1} - s_k$ was approximated by doing Gaussian elimination of the linear equation (2.46) while dropping higher order terms in Δ.

Some of the methods just mentioned will be revisited in Chapter 4. For instance, we show that MSLP can be elegantly formulated as a Newton-type *set-valued fixed point iteration*, which makes it possible to give an explicit expression for the convergence rate in terms of eigenvectors.

Finally, the *self-consistent field iteration* (SCF) (e.g. [LBLK06]) is also a sequence of linear problems, where the higher order derivatives are neglected. Some convergence properties of SCF will also be discussed in Chapter 4.

Nonlinear Rayleigh iteration

We now sketch a motivation for the Rayleigh-type iteration presented in [Wer70] (see also [Vos04b]) where it is referred to as *safeguarded iteration*. See [Rog64], [Had67] and references in [Ruh73] for other methods based on Rayleigh functionals. Here, we present the method in an informal way. See [Vos04b] for a more formal discussion and implementational details.

Suppose that we are looking for real eigenvalues within some interval I. For a solution $s_* \in I$ of the nonlinear eigenvalue problem (2.45) with corresponding eigenvector v, it holds that

$$v^* T(s_*) v = 0.$$

Let $p(v)$ be the inverse of this expression, i.e.,

$$p(v) := \{s \in I : v^* T(s) v = 0\}, \tag{2.48}$$

which is called the nonlinear Rayleigh functional. Now consider the iteration:

$$\theta \in \sigma(T(p(v_k))), \text{ with eigenvector } v_{k+1} \in \mathbb{C}^n. \tag{2.49}$$

We must choose an eigenpair from $\sigma(T(p(v_k)))$ in each step of the set-valued fixed point iteration (2.49). Assume $T(s)$ is Hermitian for all $s \in I$, then the eigenvalues are real and can be numbered by decreasing real part. If we choose the kth eigenvalue from T in each step, the method has a (surprisingly) nice convergence behaviour. In fact, under some assumptions on the interval I (such that $p(v)$ is unique) and the derivative T', the nonlinear eigenvalue problem

has a (so-called) min-max characterization and we have global convergence to the largest eigenvalue and local quadratic convergence to any eigenvalue. Now consider the improved version

$$\theta \in \sigma(T'(p(v_k))^{-1}T(p(v_k))), \text{ with eigenvector } v_{k+1} \in \mathbb{C}^n, \tag{2.50}$$

where (as usual) the corresponding generalized eigenvalue problem is solved such that the inverse must not be explicitly computed. This variant has even cubic local convergence ([Vos04b]).

Note that $p(v)$ is a scalar equation and can often be solved numerically with Newton iteration (if a good starting guess is available). Moreover, for many nonlinear eigenvalue problems T, the corresponding scalar problem $v^*T(s)v = 0$ can even be solved explicitly, e.g. the single delay scalar eigenvalue problem can be expressed with the Lambert W-function (see Section 2.2.1).

The iteration (2.50) is not suitable for very large eigenvalue problems as it requires the solution of an eigenvalue problem in each step. However, the global convergence properties makes it very suitable to be combined with projection methods (see Section 2.3.2). The method has been successfully used in combination with a projection method [Vos04a] which is essentially Algorithm 2.1. The method in [Vos04a] has been applied to problems with a min-max characterizations from fluid-solid vibration interaction [Vos03] and electronic states of quantum dots [BV07] (see also [Bet07a]).

2.3.2 Projecton methods

Vector-valued Newton iteration

The nonlinear eigenvalue problem (2.45) can be written as a system of nonlinear equations by introducing an additional normalization constraint $c^*v = 1$ for some normalization vector c. That is,

$$\begin{pmatrix} T(s)v \\ c^*v - 1 \end{pmatrix} = 0. \tag{2.51}$$

Note that, we apply the Newton iteration with approximations of the eigenvalue $s_k \in \mathbb{C}$ and eigenvector $v_k \in \mathbb{C}^n$. The Newton iteration applied to (2.51) is

$$\begin{pmatrix} v_{k+1} \\ s_{k+1} \end{pmatrix} = \begin{pmatrix} v_k \\ s_k \end{pmatrix} - \begin{pmatrix} T(s_k) & T'(s_k)v_k \\ c^* & 0 \end{pmatrix}^{-1} \begin{pmatrix} T(s_k)v_k \\ c^*v_k - 1 \end{pmatrix},$$

or

$$v_{k+1} = \frac{1}{c^*T^{-1}(s_k)T'(s_k)v_k} T^{-1}(s_k)T'(s_k)v_k, \tag{2.52}$$

$$s_{k+1} = s_k + \frac{1}{c^*T^{-1}(s_k)T'(s_k)v_k}. \tag{2.53}$$

In the literature and in implementations, it is common to introduce the intermediate vector $u_{k+1} = T^{-1}(s_k)T'(s_k)v_k$ such that the iteration is

$$\begin{aligned} u_{k+1} &= T^{-1}(s_k)T'(s_k)v_k, \\ v_{k+1} &= u_{k+1}/c^*u_{k+1}, \\ s_{k+1} &= s_k - \frac{1}{c^*u_{k+1}} = s_k - \frac{u_{k+1}^*v_{k+1}}{u_{k+1}^*u_{k+1}}. \end{aligned}$$

This is a nonlinear version of inverse iteration. This application of Newton-iteration was first used on nonlinear eigenvalue problems in [Ung50]. The equivalence to inverse iteration was clarified in [PW79].

This is indeed a competitive method for problems where it is cheap to solve the linear system $T(s)x = b$. In fact, the tool DDE-BIFTOOL [Eng00] uses this iteration to get eigenvalues with machine precision.

Note that in the iteration (2.52) a system of linear equations $T(s_k)^{-1}b$, where $b = T'(s_k)v_k$, must be solved. This linear system depends on s_k. The corresponding iteration for constant s_k will converge to the solution of a different problem. This unfortunate fact is a motivation why we will now discuss a method which has a lower convergence order, but where the matrix in the linear system is constant throughout the iteration.

Residual inverse iteration

The nonlinear version of inverse iteration

$$\begin{aligned} u_{k+1} &= T(s_k)^{-1}T'(s_k)v_k, \\ v_{k+1} &= u_{k+1}/c^*u_{k+1}, \qquad s_{k+1} = s_k - 1/c^*u_{k+1} \end{aligned}$$

can be derived from Newton's method as above. We mentioned above that an unfortunate property of this iteration is that a different linear system $T(s_k)z = b$ must be solved in each step. Since the matrix $T(s_k)$ changes in each iteration, it

is in general not possible to do a precomputation (say LU-decomposition) such that solving the system with different right-hand sides can be executed efficiently. This makes the method unsuitable for problems where it is computationally demanding to solve this system, e.g. for very large problems. Note that if s_k in $T(s_k)^{-1}$ is kept constant the iteration is unusable as it converges to the solution of a different problem. The method *residual inverse iteration* (RII) by Neumaier in [Neu85] was motivated by this fact. RII is an iterative method similar to inverse iteration where the shift can be kept constant at the cost of convergence order. Similar to the Rayleigh-iterations a scalar nonlinear problem must be solved in each step.

We now describe some theoretical properties of the iteration without paying attention to efficiency and refer the reader to the original work [Neu85] for implementational details. In exact arithmetic, one step of RII is

$$u_{k+1} = u_k - T(\sigma)^{-1}T(p(u_k))u_k \tag{2.54}$$

where $p(u_k)$ is the Rayleigh-functional in (2.48), i.e., the solution s of $u_k^* T(s)u_k = 0$. Here $\sigma \in \mathbb{C}$ denotes the shift. Note that $T(\sigma)$ is constant throughout the iteration, which allows us to make an LU-decomposition at the beginning of the iteration, and solve the triangular system for different right-hand sides efficiently. However, we have the choice to update σ if necessary. This is a computational tradeoff; if σ is kept constant, the convergence is linear (to simple eigenvalues), whereas updating $\sigma = p(u_k)$ in each step gives quadratic convergence but for each step $T(\sigma)$ is changed and we have the same computational difficulties as in inverse iteration. Roughly speaking, we wish to update σ often enough to have fast convergence, but not so often that the LU-decomposition becomes the dominating part of the iteration.

Up until now we have only considered $p(u_k)$ defined as (2.48). Neumaier points out that for RII (2.48) only makes sense if $T(s)$ is Hermitian and s is real, otherwise

$$p(u_k) := \{s \in \mathbb{C} : c^* T(\sigma)^{-1}T(s)u_k\}, \tag{2.55}$$

should be used. In the next section we will discuss projection methods which are motivated by RII.

Projection methods

Nonlinear eigenvalue problems of small or moderate dimension can often be solved reliably and efficiently. For instance, the small delay-eigenvalue problem can be solved to high accuracy by the methods discussed in Section 2.2, i.e., SOD and IGD, and polynomial and rational eigenvalue problems of moderate dimension can be solved with companion linearizations. Moreover, general nonlinear eigenvalue problems where there is a min-max-characterization can be solved reliably with the Rayleigh-type iteration (2.49) (safeguarded iteration).

For large eigenvalue problems we can use some of the iterative methods suggested in Section 2.3.1, e.g. variants of inverse iterations and Rayleigh iterations. The main drawback of these methods is that they typically only have good local convergence properties, i.e., it is necessary to have a fairly good starting guess to have reliable convergence.

The motivation for the introduction of projection methods for nonlinear eigenvalue problems can be seen as an attempt to combine the nice global convergence properties we have for methods for small problems and the good local properties of the iterative methods.

The goal of the remaining part of this section is two-fold. We wish to discuss the idea and motivation of projection methods and in particular the proposed method (Algorithm 2.1) and present technical details of the proposed method in order to make the numerical results reproducable.

The general idea of projection methods for nonlinear as well as linear eigenvalue problems is that a corresponding small projected problem contains information relevant for the large problem. More precisely, suppose $\mathcal{V}$ is a subspace and the columns in $V \in \mathbb{R}^{n\times m}$ form an orthonormal basis of $\mathcal{V}$. The idea is that the solutions of the projected problem $\det(V^*T(s)V) = 0$ contains information relevant for the larger problem $\det(T(s)) = 0$.

Several projection methods for the (linear) eigenvalue problem have been generalized to the nonlinear eigenvalue problem, e.g. the subspace acceleration of residual inverse iteration [Vos04a], [Mee01], [LBLK06] as well as nonlinear versions of Jacobi-Davidson [SBFvdV96], [BV04], [SS06].

We will focus on subspace accelerations of the residual inverse iteration (RII). Two related methods in the literature are motivated by the residual inverse it-

eration. The method by H. Voss [Vos04a] is called *nonlinear Arnoldi* and the *quadratic residual iteration* by [Mee01, Algorithm 2.3]. A similarly motivated method is called *nonlinear Rayleigh-Ritz iterative method* in [LBLK06].

Before presenting the variant used for the numerical examples, we wish to point out some differences between the works of Voss and Meerbergen. Even though the methods are equivalent both works use and point out important properties of the method. An important contribution in the work of Meerbergen is the connection to Jacobi-Davidson. A Jacobi-Davidson method is also suggested. The connection is studied by considering the Cayley-transformed problem. On the other hand, Voss shows how convergence results for the residual inverse iteration can be used in a heuristic to determine when the convergence is slow. If the convergence is slow the shift should be updated.

Both methods solve the projected problem corresponding to the solution of the Rayleigh function (2.48). That is, instead of solving the scalar problem (2.48) a corresponding projected problem is solved

$$V^*T(s)Vy = 0,$$

where V is a basis of the subspace of the current iteration. One of the eigenvectors from the projected problem is chosen and the subspace is expanded with the direction of one step of residual inverse iteration (2.54). In [Vos04a] the projected problem is suggested to be solved with inverse iteration and safeguarded iteration (and a specific strategy for adapted for the examples considered was given in [Mee01]). Moreover, the method of Meerbergen is (as the name *quadratic residual inverse iteration* indicates) adapted for the quadratic eigenvalue problem.

Meerbergen applies the projection method to problems with nonreal eigenvalues and Voss does so in [Vos06b]. Note that for nonreal eigenvalues, the motivation for the expansion direction is slightly more heuristic. According to Neumaier, "Formula [(2.48)] is appropriate only when $A(\lambda)$ is Hermitian and λ, is real; otherwise [(2.55)] has to be used". Hence, an expansion in the direction of the solution of (2.48) is not recommended. Similar points were made by Voss. Fortunately, the method still works efficiently. The method was successfully used to find complex eigenvalues efficiently in [Mee01]. Note that it is not clear how the projected problem corresponding (2.55) should be stated.

Note that many relevant problems are symmetric (or Hermitian) and have real eigenvalues [Bet07a].

In this work we will focus on nonlinear Arnoldi which we will abbreviate with SRII (subspace accelerated residual inverse iteration). SRII is given in the Algorithm 2.1, and works roughly as follows. It consists of two nested loops. The method determines eigenvalues one after another in the outer loop (Step 3-Step 13). The inner loop (Step 4-Step 10) is performed until sufficient accuracy of the current wanted eigenvalue is achieved. In each iteration of the inner loop, the considered subspace is expanded by one direction, by adding one vector to the basis. This vector is the orthogonal component of the vector result from Step 7. In Step 7 we perform one step of residual inverse iteration by using the solution of the projected problem computed in Step 6. Once convergence to an eigenvalue is achieved, the eigenvalue is stored and the subspace is reduced (Step 12) by some strategy which must be adapted to the problem at hand.

Note that each expansion of the subspace "corresponds" to one step residual inverse iteration and the convergence is expected to be at least as good as RII. Formal proofs for such statements are not available in the literature. This is unfortunately often the case for many subspace accelerations.

We now discuss details for our implementation.

Step 1 The factorization of $T(\sigma)$ should be done such that the linear system $T(\sigma)x = b$ can be solved efficiently (in Step 7). Here we do an LU-factorization with row and column reordering using the function LU provided in Matlab.

Step 2 We generate a random vector b and pick the following initial starting subspace, $V = [(T(\sigma)^{-1})b, \cdots, T(\sigma)^{-1})^5 b]$, and orthogonalize.

Step 3 Here $m_{\max}$ is the number of wanted eigenvalues.

Step 4 The relative residual is computed as $\|T(\mu)v\|/\|T(\sigma)\|$.

Step 5 The list of projected matrices *Aproj* is here updated or recomputed if necessary. This is necessary if the projection space was changed in Step 12. Otherwise the update can be done by adding the appropriate row and column.

Step 6 The function projectedsolve computes the "best" solution of the projected problem $V^*T(\mu)Vy = 0$. Which solution candidate is the best solution, is determined from some selection strategy, typically depending on the shift

Algorithm 2.1 Subspace accelerated residual inverse (SRII) (NLARNOLDI)

INPUT: $T(s) = A_0 f_0(s) + \cdots A_N f_N(s)$, shift σ, m_{max}, RESTOL, RESTARTDIM, projectedsolve$(\cdot)$, restart$(\cdot)$

OUTPUT: Eigenvalues μ_l with corresponding eigenvectors V_l

1: Factorize $LU = PT(\sigma)Q$
2: Compute initial vectors V
3: for $m = 1 \dots m_{\max}$ do
4: while $relres > RESTOL$ do
5: Update $Aproj$ such that $Aproj(k) = V^* A_k V$ for all k.
6: Solve projected nonlinear eigenvalue problem

$$[\mu, y, \mu_{next}, y_{next}] = \text{projectedsolve}(V, Aproj, \mu, \sigma, \sigma_t, V_l, \mu_l)$$

where (μ, y) is the best solution candidate from selection strategy.
7: Compute $v = T(\sigma)^{-1} T(\mu) V y$ with factorization of $T(\sigma)$.
8: Orthogonalize v against V
9: Expand search space $V = [V, v/\|v\|]$
10: end while
11: Store locked values $V_l = [V_l, v]$, $\mu_l = [\mu_l, \mu]$.
12: If dimension $V >$ RESTARTDIM restart:

$$[V, \mu] = \text{restart}(V, V_l, \mu, y, \mu_{next}, y_{next})$$

13: end for

σ, the target σ_t, the previous iterate μ and the locked eigenvalues μ_l, i.e., those eigenvalues which are already found, with corresponding eigenvectors V_l. This strategy and method must be adapted to the problem at hand. The general properties a good strategy should have is given below. The third and fourth output parameter μ_{next} and y_{next} correspond to the second best candidate or possible a list of candidates, which can be used when restarting the method (Step 12).

Step 7 The expansion direction v is taken as one step of residual inverse iteration. As usual, the inverse $T(\sigma)$ is not explicitly computed. The factorization in Step 1 can be used to solve the linear system.

Step 8 The orthogonalization is computed by $v = v - V(V^* v)$. We use reorthog-

onalization in order to avoid numerical that the space is not numerically orthogonal.

Step 11 The found eigenvector is stored by adding it as a column to the matrix V_l. The vector is stored in order to make sure the iteration does not converge to the same eigenvector several times.

Step 12 Solving the projected nonlinear eigenvalue problem (Step 6) and the orthogonalization (Step 8) can turn computationally demanding if the dimension of the search space V is large. For this reason it is computationally advantageous to reduce the subspace at some point. This restart strategy is dependent on the problem. Typically (as done in e.g. [Vos07]) the restart is done with the locked eigenvectors V_l and the second best candidate of the projected nonlinear eigenvalue problem computed in Step 6, i.e., we let $V = [V_l, Vy_{next}]$ and orthogonalize V.

If we want to find more than one eigenvalue, a critical problem for projection methods is to determine which solution to the projected eigenvalue problem should be selected. For the linear eigenvalue problem the eigenvalue is chosen closest to some target, see e.g. [SvdV96].

The situation for nonlinear eigenvalue problems is somewhat different. Unlike linear eigenvalue problems we should try to have convergence to one eigenvalue at a time, since we must insert the eigenvalue in the parameter dependent matrix $T(s)$.

A good strategy must handle the following issues.

- It must avoid convergence to locked eigenvalues, i.e., eigenvalues marked as already found.
- It converges in a monotone way to one eigenvalue at a time. It is inefficient to have convergence to several eigenvalues at a time.
- It converges first to the eigenvalues specified by some target, e.g. the shift σ or ∞.

The heuristic strategy must be carefully constructed for the problem at hand. We will combine three strategies. The strategies address the first point, i.e., the avoidance of convergence to solutions which have already converged, in the

same way. We wish to determine the eigenvalues one after another. Hence, once an eigenvalue is converged (marked as locked in Step 11), this eigenvalue must not be returned in Step 6. We will avoid converging to locked eigenvalues by removing candidate eigenvalues of the projected eigenvalue problem which has an distance less than $LOCKTOL$ to the locked eigenvalues. One eigenvector may correspond to multiple eigenvalues. Hence, it is (strictly speaking) not sufficient to check the angle between the eigenvectors. When determining if a solution of the projected eigenvalue problem is a locked eigenvalue we must also check if the approximate eigenvalue is also close to the locked eigenvalue. This extreme case has not occurred in our examples.

2.4 Numerical examples

With the following two examples we wish to show that there are cases where projection methods are considerably more efficient than methods for the DDE, i.e., SOD and IGD.

The first example (Section 2.4.1) is a problem with symmetric tridiagonal system matrices for which the real eigenvalues should be computed. The projection method turns out to be very efficient for this problem. The general purpose package DDE-BIFTOOL can solve problems of size 100, adapted version of SOD and IGD can solve problems of dimension 1000, whereas the projection method can solve problems of dimension 10^6 (one million) in a matter of minutes.

The first example is very special. For that reason we also consider a second example (Section 2.4.2) with randomly generated matrices. Here, the projection method is still superior to the IGD- and SOD-methods, but not with such a large difference as the first example.

The numerical examples are performed on a dual core AMD Athlon 64 4400+ with 2 Gb memory using Matlab 7.4.0 (R2007).

2.4.1 A PDDE with real eigenvalues

In the first example we will compute the real eigenvalues of a partial differential equation with delay (PDDE). Consider

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + a_0(x)u + a_1(x)u(x,t-\tau_1) + a_2(x)u(x,t-\tau_2), \\ \qquad\qquad u(0,t) = u(\pi,t) = 0,\ t \geq 0 \end{cases} \tag{2.56}$$

where $a_0(x) = a_0 + \alpha_0 \sin(x)$, $a_1(x) = a_1 + \alpha_1 x(1 - e^{x-\pi})$ and $a_2(x) = a_2 + \alpha_2 x(\pi - x)$. We let $u = u(x,t)$ for notational convenience. This equation is a generalization of [Wu96, Example 1.12, Chapter 3]. See also Example 3.23 for further theoretical properties and derivation of this equation.

We discretize the PDDE with central difference and uniform step-size $h =$

$\pi/(n+1)$ in space and get the DDE of dimension n

$$\dot{v}(t) = \frac{(n+1)^2}{\pi^2}\begin{pmatrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{pmatrix} v(t) + \begin{pmatrix} a_0(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_0(x_n) \end{pmatrix} v(t) + \begin{pmatrix} a_1(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_1(x_n) \end{pmatrix} v(t-\tau_1) + \begin{pmatrix} a_2(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_2(x_n) \end{pmatrix} v(t-\tau_2). \quad (2.57)$$

Here we will let $\tau = \tau_1 = \tau_2$, but in later chapters this example will be revisited with different delays. We select $a_0 = 20, \alpha_0 = 0, a_1 = -4, \alpha_1 = 1, a_2 = -0.1$, $\alpha_2 = 0$ and $\tau = 0.05$.

For $\tau = 0$ the PDE (as well its discretization) has only real eigenvalues. For small delays several eigenvalues will remain real. We first try some SOD-methods, IGD-methods and the projection method to find the 12 rightmost of these real eigenvalues.

We apply the methods to the DDE (2.57). The timing-results for different discretization sizes n are shown in Table 2.4. We are able to solve problems of dimension $n = 10^6$ using SRII. Note that we have ignored the experiments for $n = 10^5$ for typographical reasons.

The column with header *nof. eigs.* denotes the number of real eigenvalues which were found to an accuracy 10^{-1}. We have chosen such a high tolerance for this indicator in order to show that the parameter N in the IGD- and the SOD-methods can not be decreased without missing some eigenvalues. The column marked as *CPU* denotes the (wall-clock) CPU time-consumption. The entries marked as MEMERR correspond to runs where either Matlab (or in particular eigs) threw an out of memory exception, or the primary memory was filled such that excessive swapping caused the method to slow down very much.

We propose the following interpretations of Table 2.4. DDE-BIFTOOL is general purpose package which uses an implementation of the SOD(MS) (as default) and full matrices. It can currently not handle sparse matrices. Hence, it is not surprising that it does not work well for larger (sparse) problems. However, our implementation of the SOD(MS) which uses the sparse structure also performs comparatively bad. This indicates that even if DDE-BIFTOOL would be modified to handle sparse matrices, it would still not handle this problem well (at

Method	$n = 100$		$n = 10^3$		$n = 10^4$		$n = 10^6$	
	nof. eigs	CPU	nof. eigs	CPU	nof. eigs	CPU	nof. eigs	CPU
DDE-BIFTOOL 2.03 minrealpart=-20	5	18.0s	MEMERR					
SOD(MS,$N = 4$)	6	0.2s	6	3.3s	MEMERR			
SOD(MS,$N = 8$)	10	0.4s	9	6.1s	MEMERR			
IGD(PS,$N = 5$)	5	0.3s	5	2.6s	MEMERR			
IGD(PS,$N = 10$)	12	0.4s	12	27.4s	MEMERR			
IGD(Euler, $N = 10$)	4	0.1s	4	1.2s	MEMERR			
IGD(Euler, $N = 100$)	5	0.9s	5	12.9s	MEMERR			
SRII(RI) restart=0	12	1.3s	12	0.5s	12	2.8s	12	361.7s
SRII(RI) restart=20	12	0.3s	12	0.3s	12	1.9s	12	306.5s
SRII(RI) restart=∞	12	0.2s	12	0.3s	12	1.9s	12	292.5s

Table 2.4: The methods applied to (2.57). Abbreviations: SOD=Solution operator discretization (Section 2.2.2), MS=Milne-Simpson, IGD=Infinitesimal generator discretization (Section 2.2.3), PS=Pseudospectral approximation, RI=Rayleigh iteration (2.49) (safeguarded iteration), SRII=Subspace accelerated residual inverse iteration, i.e., NLARNOLDI, Algorithm 2.1.

least for larger n). The memory and not computational power is the threshold for all IGD and SOD-methods. We have shown that some of the methods are linearizations of polynomial eigenvalue problems. Linearizations of polynomials are known to be a memory consuming way to solve polynomial eigenvalue problems, as any linearization increases the dimension of the matrix by a factor equal to the order of the polynomial minus one.

There are 12 eigenvalues in the interval $[-70, 20]$. Apart from the projection method, only a fine discretization with IGD(PS) could determine all of them to the (very coarse) accuracy 0.1. This stems from the fact that the methods are approximations or interpolations of the DDE at certain points, e.g. the origin. It is not surprising that the eigenvalues far away from the origin are badly approximated. This can also been seen in Table 2.5. In Table 2.5 the error is computed comparing the approximated solution with a more accurate solution computed with some steps of Newton iteration.

We made three experiments with SRII for different restart strategies. If the subspace has grown to a dimension larger than the parameter $RESTARTDIM$ the method will be restarted. The method was restarted with (the locked space) and an eigenvector of the delay-free system corresponding to the same eigenvalue.

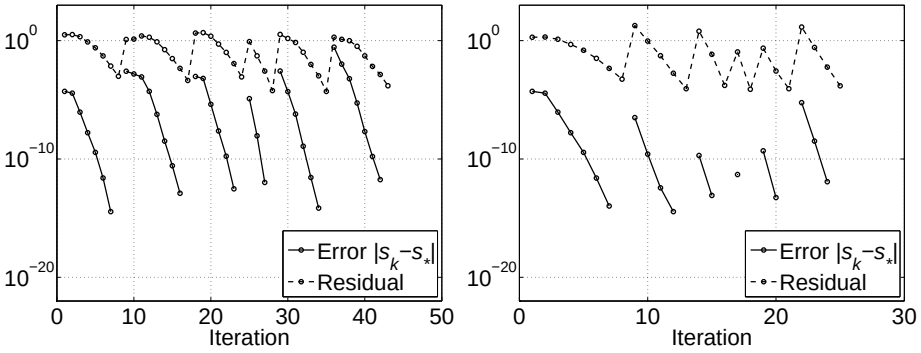
$n = 10^6$	$n = 10^3$	
SRII(RI)	SRII(RI)	IGD(PS, $N = 10$) Error
17.77390548351727	17.77390636019264	3.62e-10
14.47147628813390	14.47149051585005	3.26e-09
8.96126129910668	8.96133538774918	1.74e-10
0.94108839008011	0.94133654158960	9.17e-09
-10.40800890309490	-10.40730527562455	1.20e-09
-31.76066692537303	-31.75561552146499	2.37e-08
-40.85518094704453	-40.85952887199092	9.81e-07
-54.65155800120748	-54.65191370323936	8.95e-06
-59.62703593778813	-59.62714883199634	2.17e-05
-62.90419784708817	-62.90424316387534	3.88e-05
-65.73788326232723	-65.73790047830707	6.37e-05
-68.46274387212003	-68.46274748864577	1.01e-04

Table 2.5: The eigenvalues and error for some of the runs. The error for IGD(PS) is larger for eigenvalues far away from the origin.

In order to make the result reproducable, we will now present technical implementational details.

For all IGD and SOD methods we used `eigs` as eigenvalue solver. We set the parameter such that 20 eigenvalues were found with target 0. The more natural target for this problem would be the smallest imaginary part. The experiments with this target performed worse because the approximated eigenvalue jumped between different approximations.

We will now describe the problem specific details for SRII, i.e., Algorithm 2.1. We use the nonlinear Rayleigh iteration (2.49) (safeguarded iteration) as a solver for the projected problem. We terminate the Rayleigh iteration when $|s_k - s_{k+1}| < 10^{-10}$. Rayleigh iteration (safeguarded iteration) only works well when p is unique, which is not the case for the interval containing the 12 rightmost eigenvalues. We resolve this by separating the interval into two parts $I_1 = [-40, 20]$ and $I_2 = [-70, -40]$. We start the iteration with the 3 of the eigenvectors of the delay free system, corresponding to the rightmost eigenvalues. As mentioned



(a) restart=0 (b) restart=∞

Figure 2.5: Iteration history for SRII in interval $[-40, 20]$ and $N = 10^6$. The eigenvalues in the middle converge fast because they are close to the shift.

the 10 first eigenvectors of the delay-free system are also used when restarting. Note that Rayleigh iteration (safeguarded iteration) is very suitable for problems with real eigenvalues because, if there is a min-max numbering of the eigenvalues, one can use the eigenvalue number as a target in the strategy for the projected problem. This is done here.

SRII turned out to be successful and superior to the other methods for this example. Note that SRII can be improved by constructing a shift update strategy or more sophisticated restart strategies. Moreover, for a formally correct use of safeguarded iteration, a min-max characterization of (2.56) should be proven. See [Vos04a] for further enhancements.

We now wish to point out some shortcomings of the construction of this example, motivating the study of a different example in the next section. The system matrices are tridiagonal and symmetric and the delay is small. This is a very special class of problems which is likely easier to solve than problems with general matrices and delays. In particular, this example can probably be solved directly with a fixed point method using the corresponding delay-free vector as initial guess.

If one is interested in the eigenvalues of the PDE (2.56), a more sophisticated discretization scheme than central difference is likely to be more efficient.

2.4.2 Random matrix with complex eigenvalues

The discretization of the PDE with delay in the previous example resulted in matrices with diagonal and tridiagonal structure. Since this simple structure is typically not present in more realistic applications, we wish to consider another example with large and sparse matrices, but without the strong diagonal structure.

We consider a random matrix constructed by the following Matlab-commands:

```
rand('seed',0); randn('seed',0);
A0=alpha*sprandn(n,n,beta); A1=alpha*sprandn(n,n,beta);
```

That is, $A_0, A_1 \in \mathbb{R}^{n\times n}$ are sparse matrices with sparsity density β and random normally distributed numbers with standard deviation α. The random seed is reset in order to have reproducable results between different runs (but unfortunately not between different system setups).

In the tests below we have chosen the scaling α such that we have approximately 10 eigenvalues with magnitude less than one half, i.e., $|s| \leq 0.5$. We now investigate with what efficiency and to what accuracy (absolute error) the methods can compute these eigenvalues. The accuracy and timing results for some choices of n, α and β are shown in Table 2.6. The CPU-columns in the table contain the (wall-clock) timing and the accuracy is the maximum error in the computed eigenvalues, where we have performed two steps of inverse iteration to have a (hopefully) considerably more accurate approximation to compare with.

It is clear from the table that for small and moderately sized problems the pseudo-spectral discretization of the PDE, i.e., IGD(PS) is fast and accurate. For dimensions larger than $n = 4000$ the projection method is the only method which can solve the problem.

We now make some notes on the implementational details for the methods. The implementation of SOD (with Milne-Simpson discretization) makes a companion linearization resulting in a generalized eigenvalue problem of dimension $(N + 1)n$. This generalized eigenvalue problem is solved using `eigs` (an implementation of *implicitly restarted Arnoldi method*) with target $\sigma = 1$, as $\mu = 1$ corresponds to $s = 0$. The 15 eigenvalues closest to $\sigma = 1$ are found. This is chosen larger than the number of wanted eigenvalues (10) as the k eigenvalues closest to $\mu = 1$ do not necessarily transform to the k eigenvalues closest to

Method	$n = 100$ $\alpha = 0.7$ $\beta = 0.1$		$n = 1000$ $\alpha = 6$ $\beta = 0.1$		$n = 2000$ $\alpha = 9$ $\beta = 0.1$		$n = 4000$ $\alpha = 10$ $\beta = 0.05$		$n = 10000$ $\alpha = 15$ $\beta = 0.001$	
	Accuracy	CPU	Accuracy	CPU	Accuracy	CPU	Accuracy	CPU	Accuracy	CPU
SOD(MS, $N = 2$)	4.8e-05	0.3s	8.4e-06	19.3s	8.5e-06	109.7s	MEMERR			
SOD(MS, $N = 4$)	6.1e-07	0.3s	1.0e-07	38.5s	1.0e-07	193.2s	MEMERR			
IGD(PS, $N = 3$)	6.3e-03	0.3s	5.5e-03	9.8s	7.7e-03	47.8s	4.9e-03	321.5s		
IGD(PS, $N = 5$)	2.8e-06	0.1s	2.3e-06	11.7s	3.7e-06	105.4s	MEMERR			
IGD(Euler, $N = 10$)	1.0e-02	0.6s	1.1e-02	131.1s	1.2e-02	399.0s	MEMERR			
IGD(Euler, $N = 100$)	1.0e-03	5.4s	1.1e-03	680.8s	MEMERR					
SRII(IGD) restartdim=0	3.8e-04	14.0s	2.6e-05	55.3s	1.0e-04	69.3s	1.1e-04	225.0s	1.1e-03	523.0s
SRII(IGD) restartdim=50	3.8e-04	29.9s	4.5e-05	91.2s	1.1e-04	75.3s	4.6e-05	245.5s	2.0e-04	521.4s
SRII(IGD) restartdim=∞	3.8e-04	38.1s	4.5e-05	113.5s	1.1e-04	91.0s	2.3e-01	286.3s	2.0e-04	518.7s

Table 2.6: CPU-time of for the random matrix. Abbreviations: SOD=Solution operator discretization (Section 2.2.2), MS=Milne-Simpson, IGD=Infinitesimal generator discretization (Section 2.2.3), PS=Pseudospectral approximation. SRII=Subspace accelerated residual inverse iteration, i.e., the variant of [Vos04a] in Algorithm 2.1

$s = 0$. The matrix-vector product provided to `eigs` involves the solving of a linear system, which must be solved many times for different right-hand sides. The standard procedure for this type of problem is to do an LU-decomposition of the shifted matrix before starting the iteration, and perform the backward and forward substitutions in every matrix vector product. This is done with partial pivoting and improves the efficiency considerably.

For IGD we also use `eigs` on the (sparse) matrix approximating the infinitesimal generator. The dimension of the constructed matrix is of size Nn. The target here is chosen to be $\sigma = 0$, and (to have a fair comparison) we set `eigs` to find the 15 smallest eigenvalues.

The subspace accelerated residual inverse is performed to the residual toler-

ance 10^{-2}. The projected eigenvalue problem is solved with SOD(PS,$N = 6$). In the solver for the projected system, we use `eig` for matrices of dimension smaller than 150 and `eigs` otherwise. We configure `eigs` to find the 30 eigenvalues closest to the origin. This value can be decreased improving the efficiency. Here, it is not necessary to fine-tune this parameter, to support the theory that the projection method is efficient for larger systems. To choose the eigenvalue out of the set of solutions to the projected problem, we use the weighted strategy suggested above with weighting $k = 4$, and $LOCKTOL = 10^{-3}$.

This example can not be handled with the normal settings of the Matlab package DDE-BIFTOOL 2.03. The package is based on SOD (with default Milne-Simpson) but unlike our tests here, it constructs *full* matrices, and applies `eig` (i.e., the Lapack QR-method) to the companion linearization. The constructed full matrices for this example are too large to be solved with `eig` in reasonable time. Even with the improved heuristic suggested in [VGR04], the choice of N will be larger than the first rows in Table 2.6 and will (even with `eigs`) be slower than the tests here. However, we note that the goal of the heuristic for N is to give sufficient accuracy on the rightmost eigenvalues, whereas we here search for the eigenvalues closest to the origin.

Chapter 3

Critical Delays

The asymptotic behavior of DDEs is an important property in many applications. We call a DDE *asymptotically stable* if the solution $x(t)$ is bounded and goes to zero as time goes to infinity, for any bounded initial condition[1]. The general problem considered in this chapter is to determine sufficient and necessary conditions for asymptotic stability. The asymptotic stability of a DDE can be established from the rightmost part of the spectrum. In particular, if the supremum of the real part of the spectrum is negative, then the DDE is asymptotically stable. The supremum of the real part of the spectrum is called the *spectral abcsissa*.

In the previous chapter we discussed numerical methods to compute the spectrum of DDEs with a single delay, i.e., the derivative of the state was a linear combination of the state and a delayed state. We now consider more general classes of DDEs. Most results are applicable to DDEs with an arbitrary number of delays. Moreover, we present results for *neutral DDEs*, i.e., DDEs containing terms with delays in the derivative of the state. In particular we discuss stability properties as a function of the delays.

For most cases, the following phenomena will occur at a point where the stability of the DDE changes. Consider a curve between two points in the space of the delay parameters (called *delay-space*). Suppose that the DDE is stable for one point and unstable for the other. When varying the delay parameters along this curve, the spectrum changes continuously (in some sense), and at the point

[1] This is the usual definition of stability in the context of time-delay systems.

where the stability of the DDE changes, one eigenvalue crosses the imaginary axis. Hence, for most cases, the boundary of the points in delay-space for which the DDE is stable is a subset of the set of points where the DDE has a purely imaginary eigenvalue. We call the set of points where the DDE has a purely imaginary eigenvalue, the set of *critical delays*.

The goal of this chapter is to compute or *parameterize* such delays with explicit expressions or numerical procedures.

We describe the phenomena in more precise terms. If the spectral abscissa is continuous with respect to changes in the delays, and the origin is not a clustering point of the real part of the spectrum, then there is a purely imaginary eigenvalue at a point (in delay-space) where the DDE goes from stable to unstable. Fortunately, the spectral abscissa is continuous for most interesting cases and the case that the origin is a clustering point of the real part is a degenerate case which is easy to identify. Under these conditions, we characterize the boundary of the set of points in delay-parameter space for which the DDE is stable as follows. This set of points is a subset of the set of delay-parameters such that there is a purely imaginary eigenvalue, i.e., the set of critical delays. A characterization of the set of critical delays is often used in methods to analyze delay-dependent stability for DDEs.

As mentioned, the main contribution of this chapter is an attempt to produce an exact characterization of the set of critical delays. In our context, by exact characterization, we mean a closed explicit formula containing only elementary functions or operations elementary in the sense that the evaluation can be performed with mature numerical procedures.

The exact representation we present is a parameterization of the set of critical delays. By this, we mean a mapping from a simple mathematical object onto the set of all subsets of the critical delays. Moreover it has the surjective property that the union of the range is exactly the set of critical delays.

The parameterization is constructed in such a way that an evaluation of the mapping consists of solving an eigenvalue problem for the general case (or more precisely of solving a *quadratic eigenvalue problem*). Moreover, the parameterization can be expressed in terms of trigonometric expressions for the case that the DDE is scalar.

Apart from constructing such a parameterization, we show how the critical

delays of commensurate DDEs, i.e., DDEs where the delays are fixed integer multiples of each other, can be parameterized using (so-called) *polynomial eigenvalue problems*.

The chapter is organized into the following main parts. In the first part (Section 3.2) we review some of the large number of contributions in the field of delay-dependent stability analysis of time-delay systems, in particular the contributions related to imaginary eigenvalues. In the second part we present a method to parameterize the critical delays for retarded (Section 3.3) as well as neutral DDEs (Section 3.4). The computationally demanding part in an evaluation of the parameterization is to find the unit magnitude eigenvalues of a quadratic eigenvalue problem. We discuss computational topics related to this *quadratic eigenvalue problem* in Section 3.5.

We also present delay-dependent results which are not parameterizations. In Section 3.6 we present an alternative way to derive the essential parts of one class of stability conditions in the literature. The class of conditions is referred to as matrix pencil methods, because the conditions are expressed as a generalized eigenvalue problem. The eigenvalue problems normally involve Kronecker products. The alternative derivation is based on a generalization of a two-parameter eigenvalue problem, which can be rewritten into a generalized eigenvalue problem in different ways corresponding to the different types of matrix pencil methods.

Finally, we discuss a result of Toker and Özbai [TÖ96] in Section 3.7. This result essentially states that a problem related to the stability of DDE belongs to the class of problems called NP-hard. We motivate why this does not necessarily imply that the problem we consider is difficult from a numerical point of view.

3.1 Introduction

In a large part of this chapter we study n-dimensional retarded delay-differential equations with m discrete delays, i.e.,

$$\Sigma = \begin{cases} \dot{x}(t) = \sum_{k=0}^{m} A_k x(t-h_k),\ t \geq 0 \\ x(t) = \varphi(t),\ t \in [-h_m, 0], \end{cases} \tag{3.1}$$

where $A_0, \ldots, A_m \in \mathbb{R}^{n\times n}$. This retarded DDE has the characteristic equation

$$\det\left(M(s)\right) = 0, \tag{3.2}$$

where

$$M(s) := -sI_n + \sum_{k=0}^{m} A_k e^{-h_k s}. \tag{3.3}$$

Without loss of generality we let $h_0 = 0$ for notational convenience. We will denote the spectrum by $\sigma(\Sigma) := \{s \in \mathbb{C} : \det(M(s)) = 0\}$ or sometimes $\sigma(\Sigma(h_1, \ldots, h_m))$ in order to stress the dependence on the delay parameters.

Several results of this chapter can be derived for a more general type of DDE. We consider DDEs with a delay in one or more derivative terms. These DDEs, known as *neutral DDEs*, are given by

$$\Sigma = \begin{cases} \sum_{k=0}^{m} B_k \dot{x}(t-h_k) = \sum_{k=0}^{m} A_k x(t-h_k),\ t \geq 0 \\ x(t) = \varphi(t),\ t \in [-h_m, 0], \end{cases} \tag{3.4}$$

where $A_0, \ldots, A_m, B_0, \ldots, B_m \in \mathbb{R}^{n\times n}$. Analogously, the characteristic equation of (3.4) is

$$\det(M(s)) := \det(-sB(s) + A(s)) = 0, \tag{3.5}$$

where $A(s) = A_0 + \sum_{k=1}^{m} A_k e^{-h_k s}$ and $B(s) = B_0 + \sum_{k=1}^{m} B_k e^{-h_k s}$. The spectrum is the set of roots of the characteristic equation, i.e., $\sigma(\Sigma) := \{s \in \mathbb{C} : \det(M(s)) = 0\}$.

It is desirable in many applications that the solution $x(t)$ goes to zero independent of the initial condition. This and many other properties are often analyzed using the spectrum $\sigma(\Sigma)$. This holds in particular, for the stability of DDEs.

We now give a short overview of stability concepts relevant in our context. See the standard references on DDEs for a more complete discussion, e.g. [HV93], [DvGVW95] or [MN07b]. The continuity results are mostly from [MN07b] and [MERD02]. For some further notes on continuity see [Pli05].

A DDE is called *asymptotically stable* if the solution is bounded and goes to zero independent of (bounded) initial condition. In fact, more accurately, the stability of a DDE is defined as the stability of the (so-called) *null solution*. The null solution of a DDE is asymptotically stable if for any given $\delta > 0$ and any DDE with initial conditions φ bounded by δ, the solution is bounded by some $\varepsilon > 0$ for all t, i.e., $|x(t)| < \varepsilon$, and $x(t) \to 0$ as $t \to \infty$ [MN07b, Definition 1.18].

In terms of the spectrum, a DDE is asymptotically stable if the real part of the spectrum $\sigma(\Sigma)$ is bounded from above by a negative number [MN07b, Proposition 1.20]. Unless zero is a clustering point of the real part of the spectrum, it is sufficient to require that the spectrum is contained in the open left half plane, i.e., $\sigma(\Sigma) \subset \mathbb{C}_-$. In particular, this holds for retarded DDEs as the real part of the spectrum of a retarded DDE has no clustering points [MN07b, Proposition 1.6]. The clustering points of the real part of the spectrum for neutral DDEs are also often non-zero.

The stability analysis we carry out consists of a continuity argument using the spectrum. Most of the continuity results available in the literature are stated in terms of the spectral abscissa of the DDE. The spectral abscissa is defined as the real part of the rightmost eigenvalue or the corresponding supremum, i.e.,

$$\alpha(h_1, \ldots, h_m) := \sup\{\text{Re } \sigma(\Sigma(h_1, \ldots, h_m))\}. \tag{3.6}$$

The asymptotic stability of the DDE is now equivalent to that the spectral abscissa is negative, i.e., $\alpha(h_1, \ldots, h_m) < 0$.

For retarded DDEs the spectral abscissa α is continuous with respect to the delays [Dat78]. For neutral DDEs, α is not always continuous (see counter example [MN07b, Example 1.38]). However, sufficient conditions on the DDE such that it is continuous are available in the literature. We postpone the discussion of continuity properties of neutral DDEs to Section 3.4.

We will focus on stability conditions in the delay parameters, to this end we define D_s as the set of the delay parameters $(h_1, \ldots, h_m)$ such that the DDE is stable, i.e.,

$$D_s := \{(h_1, \ldots, h_m) \in \mathbb{R}_+^m : \alpha(h_1, \ldots, h_m) < 0\}.$$

For retarded DDEs, the real part of the spectrum has no clustering points and $D_s = \{(h_1, \ldots, h_m) \in \mathbb{R}_+^m : \sigma(\Sigma(h_1, \ldots, h_m)) \subset \mathbb{C}_-\}$. Since α is continuous for retarded DDEs, the spectral abscissa α is zero on the boundary of the stability region ∂D_s, i.e., $\partial D_s \subset \{(h_1, \ldots, h_m) \in \mathbb{R}_+^m : \alpha(h_1, \ldots, h_m) = 0\}$.

Note that for retarded DDEs and neutral DDEs for which the origin is not a clustering point of the real part of the spectrum, assuming that the spectral abscissa is zero, i.e., $\alpha(h_1, \ldots, h_m) = 0$, implies that there is a (finite) purely imaginary eigenvalue.

In other words, if $\alpha(h_1, \ldots h_m) = 0$ then there is a purely imaginary eigenvalue, or the imaginary axis is a clustering point of the spectrum. Hence, a characterization of the points in delay-space where there is a purely imaginary eigenvalue can be useful in a stability analysis. We will denote this set of points by $\mathcal{T}$. We call $\mathcal{T}$ the set of *critical delays*, *critical curves* ($m = 2$), *critical surface* ($m = 3$) or *critical hypersurfaces* ($m > 3$). That is, the critical delays are,

$$\mathcal{T} := \{(h_1, \ldots, h_m) \in \mathbb{R}_+^m : \sigma(\Sigma(h_1, \ldots, h_m)) \cap i\mathbb{R} \neq \emptyset\}.$$

The continuity argument used in this chapter is that if α is continuous, and zero is not a clustering point of the real part of the spectrum, then $\partial D_s \subset \mathcal{T}$.

Remark 3.1 (Terminology) The set of critical delays $\mathcal{T}$ (or slight variations thereof) has several names in the literature, e.g. *stability crossing curves/boundaries* (e.g. in [MNG06], [GNC05], [NKGL06], [Mor06]) *switching delays*, *switching points* (e.g. [BK02], [Nic01a]), *Hopf bifurcation curves* (e.g. in [HH93]), *delay crossing set* (e.g in [MN07b, Chapter 4]) or *kernel* and *offspring curves* (in [SO05] and derivative works). The term *critical delay* is used in [PFF$^+$06] and [Pli05]. A slightly different meaning of *critical delays* is used to refer to the case that there is a (so-called) additional eigenvalue on the imaginary axis in [GN00]. The additional eigenvalues are eigenvalues introduced by a transformation of the time-delay system. They are additional in the sense that they are eigenvalues of the transformed system, but not necessarily of the original system.

In this work we use the term *critical delay, curves and surface* for $\mathcal{T}$. We find this terminology slightly clearer than *stability crossing curves* which is sometimes used by some experts in the field. Even though the term critical delays is less informative than stability crossing curves, we prefer it, because in some situations it is tempting to (falsely) deduce that the stability of the DDE switches (goes from

unstable to stable or vice versa) when we move across a *stability crossing curve*. Of course, the stability switches only when crossing ∂D_s, but not necessarily when crossing $\mathcal{T}$.

The terms *offspring* and *kernel curves* are avoided because the words *offspring* and *kernel* give no information about what phenomenon occurs at the points of the set $\mathcal{T}$. Moreover, "kernel" already has a precise mathematical meaning and the separation of $\mathcal{T}$ into two parts (kernel and offspring) does not ease the presentation in our context, nor does it provide further understanding.

3.1.1 Problem formulation

In the field of mathematical analysis, systems of *linear homogeneous ordinary differential equations (ODEs) with constant coefficients* are often solved by diagonalizing the system matrix, or determining the Jordan form. The solution is typically expressed as a sum of exponential or trigonometric functions of time. It is an *explicit exact representation* of the integrated ODE, from which many properties can be read off with little effort. For instance, asymptotic behaviour, maximas, minimas, oscillations, etc. can be determined by inspection or by simple trigonometric, exponential, logarithmic manipulations of the solution. The main goal of this chapter is to give an *explicit exact representation* of $\mathcal{T}$ such that relevant information can be easily determined. We will discuss ways to solve the following problem.

Problem 3.2 (The exact delay-dependent stability problem) *Find an exact representation*[2] *of $\mathcal{T}$, such that a lot of information about $\mathcal{T}$ is easily available by explicit formulas containing a finite number of elementary functions or a finite number of operations for which there are mature numerical methods.*

It is important to note that our main goal is to give an exact description of $\mathcal{T}$, i.e., not approximate, estimate, numerically compute or plot $\mathcal{T}$. This can be compared to the different ways to analyze ODEs in analysis and numerical analysis. The solution of ordinary differential equations is often numerically computed by some finite difference and diagonalization is normally not the most efficient way to numerically compute the solution of an ODE or an initial value problem. In

[2] By *exact representation* we formally mean a finite set of necessary and sufficient mathematical expressions.

the same sense, the goal of this chapter is not to produce an efficient numerical method to compute $\mathcal{T}$, but to determine a simple explicit form of $\mathcal{T}$ from which many properties can be determined by inspection or simple manipulations.

We suggest a solution to Problem 3.2 by stating a parameterization of $\mathcal{T}$. The parameterization is presented for retarded DDEs in Section 3.3 and for neutral DDEs in Section 3.4. The main result, i.e., the parameterization, is a formula containing trigonometric functions for the scalar case ($n = 1$) and an eigenvalue problem in general ($n > 1$). The parameterization for the general multidimensional case is a formula containing an eigenvalue problem, or more precisely a *quadratic eigenvalue problem*. Eigenvalue problems are one of the fundamental problems in the field of numerical linear algebra and for moderately sized eigenvalue matrices, the eigenvalues can be computed to sufficient accuracy in reasonable time.

We will of course visualize $\mathcal{T}$ in practical examples, but this is not the main goal. However, in practice, e.g. when we wish to plot $\mathcal{T}$, we must be able to numerically evaluate the parameterization. Hence, numerical efficiency must also be considered. We present some ways to improve numerical efficiency for the considered quadratic eigenvalue problem in Section 3.5. In particular, the matrix-vector product corresponding to the quadratic eigenvalue problem consists of a Lyapunov equation. The Lyapunov equation is a commonly occurring matrix equation in control theory, and there are numerical methods for solving the Lyapunov equation.

Remark 3.3 *The set of critical delays can be computed numerically in a number of ways. In fact, it is easy to think of more efficient techniques to numerically compute $\mathcal{T}$ than the parameterization we propose. For instance, one could discretize the delay-space and numerically compute some eigenvalues for each discretization point using any of the techniques in Chapter 2. This straightforward numerical approach has been used in e.g. [BMV05a]. Another natural approach is to make a grid on $\partial\mathbb{D}^m$, i.e. $z_k \in \partial\mathbb{D}$ ($\partial\mathbb{D}$ is the unit circle); substitute $z_k = e^{-ih_k\omega}$, $k = 1, \ldots, m$ and check for which values of $z_1, \ldots, z_m$, $A_0 + \sum_{k=1}^{m} A_k z_k$ has a purely imaginary eigenvalue. This has similarities with* frequency sweeping *techniques, cf. [GKC03, Section 2.3], [Nic01a, Section 3.5.1]. Subsequently, sufficient accuracy (typically machine precision ϵ_{mach}, i.e., the accuracy of the floating point system used on the computer) could be achieved applying a Newton-iteration.*

Note that neither of the techniques above qualify as solutions to Problem 3.2 as they do not provide an exact nor explicit representation of $\mathcal{T}$. In particular, the methods are no parameterizations.

The main focus of this chapter, i.e., Problem 3.2, is an instance of a wider class problems which we refer to as the *delay-dependent stability problem*. This important problem essentially consists of describing D_s, ∂D_s or $\mathcal{T}$ in some mathematical way. The following is an informal formulation of the problem to which a delay-dependent stability result should provide some information to.

Problem 3.4 (The delay-dependent stability problem)

Give information about which points in $\mathbb{R}_+^m$ belong to D_s, by determining necessary and or sufficient conditions for D_s, ∂D_s or $\mathcal{T}$.

In the discussions and examples that follow, we wish to describe in what way the informal formulation of the delay-dependent stability problem has several solutions or partial solutions. In particular, for DDEs with one single delay, this problem is (in a sense) well posed, but for DDEs with multiple delays it is not clear how a solution (or partial solution) to the delay-dependent stability problem should be stated.

For a DDE with a single delay, we can give one solution to Problem 3.4 which provides essentially all information about D_s, and could also be seen as an answer to Problem 3.2. For these DDEs, which only depend on one real parameter (the delay h), D_s is in general a countable set of intervals, and often just one interval including the origin. Hence, a numerical answer to the delay-dependent stability problem is a numerical value together with a periodicity formula. Another possible more theoretical approach is to give a formula consisting of elementary functions or operations, e.g. trigonometric functions. This is done in the following scalar example.

Example 3.5 *The set of critical delays of the single delay, scalar DDE,*

$$\dot{x}(t) = a_0 x(t) + a_1 x(t - h_1)$$

is given by,

$$\mathcal{T} = \left\{ \frac{-\operatorname{sgn}(a_1)}{\sqrt{a_1^2 - a_0^2}} \left(\operatorname{acos}\left(-\frac{a_0}{a_1} \right) + 2p\pi \right) \right\}_{p \in \mathbb{Z}}. \tag{3.7}$$

This formula is easily derived by separating the real and imaginary part of the characteristic equation (3.3)*. In Section 3.3 (in particular in equation* (3.30)*) we will show that it is a special case of the parameterization presented in this chapter. If we assume that* $0 \in D_s$*, then*

$$D_s = \left[0, \frac{1}{\sqrt{a_1^2 - a_0^2}} \operatorname{acos}\left(-\frac{a_0}{a_1} \right) \right).$$

These formulas are not new (see e.g. [Nic01a, Proposition 3.15], [GKC03, Example 2.4]). It is however interesting that the direct generalization of (3.7) *to neutral DDEs (derived in Example 3.33) is not widely known. In fact, we have only seen parts of the generalization of* (3.30)*, i.e.,* (3.56)*, in the literature.*

Unfortunately, the geometry of the sets D_s, ∂D_s and $\mathcal{T}$ is in general for multiple delays complicated. The stability region D_s is in general not convex, does not necessarily include the origin, can be unbounded, and is not even necessarily connected. For neutral DDEs, D_s can even have an empty interior whereas the closure is $\mathbb{R}^m$ (D_s is dense in $\mathbb{R}^m$). However, for most cases ∂D_s is piecewise smooth and in many applications the origin is indeed in D_s. Moreover, for DDEs with two delays, some geometric classification is available [HH93]. The difficulty in characterizing D_s, ∂D_s and $\mathcal{T}$ is possibly one justification for the large amount of literature in the field of delay-dependent stability analysis. We discuss some literature in Section 3.2.

For DDEs with multiple delays, Problem 3.4 is not very well posed in the sense that it is not clear how an answer should be formulated such that (essentially) all information about D_s is easily available. In particular, it is not clear how a generalization of the formulas in Example 3.5 should be stated such that we can identify many properties by inspection. In the example below, i.e., Example 3.6, we show that one way to generalize the formula for $\mathcal{T}$, i.e., (3.7), is a parameterization.

In the set of results available in the literature, we often have a tradeoff between information loss in the representation of the answer and explicitness of the answer. Most results are either only sufficient but not necessary conditions, or sufficient and necessary but stated in an implicit way such that it is not possible to read off a lot of information by inspection.

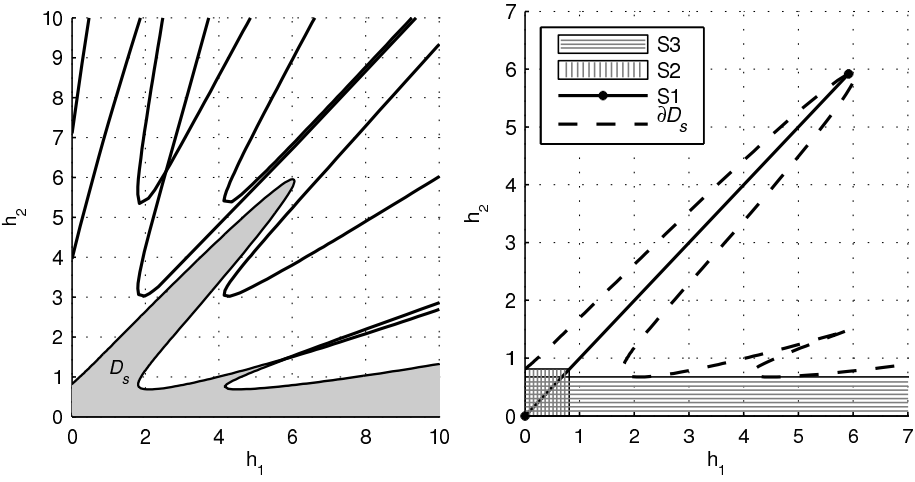
In the following example we state some ways to provide information about D_s in terms of explicit but not necessary conditions and some ways to state sufficient

and necessary by slightly more implicit statements.

Example 3.6 Consider the DDE

$$\dot{x}(t) = -x(t) + \frac{9}{10}x(t-h_1) - 2x(t-h_2), \tag{3.8}$$

which is stable for all $(h_1, h_2) \in D_s$ shown in Figure 3.1(a). This type of visualization of D_s is called a *stability chart*. It is a common tool in the literature on time-delay systems and in particular applications of time-delay systems, e.g. [HV04], [Sté89] [BC94], [CNW06], [Pee06], [AP05].



(a) Stable region D_s and critical curves $\mathcal{T}$ (b) ∂D_s and stability quantities

Figure 3.1: Stable region in delay-space

The two subfigures in Figure 3.1 show some ways to describe the stability region. We discuss the advantages and disadvantages of these ways to analyze stability next.

S1 In some applications, it is natural (e.g. for physical reasons) to assume that the delays are equal, or almost equal. One way to provide information about D_s (and hence information about Problem 3.4) is to determine what

line segment along a given ray in delay space the DDE is stable. We now establish the largest delay τ such that if $h_1 = h_2 < \tau$ then the DDE is stable. This set of stable points in delay-space is

$$\{(h_1, h_2) \in D_s : h_1 = h_2\} = \{(h_1, h_2) \in \mathbb{R}^2_+ : h_1 = h_2 < \tau\}.$$

If $h_1 = h_2$ then the DDE (3.8) reduces to a DDE with a single delay and we can use the formula for the scalar case, i.e., (3.7). Here, the line segment in delay space along the ray $h_1 = h_2$, for which the DDE is stable, is given by

$$\{(h_1, h_2) \in D_s : h_1 = h_2\} = \left\{(h_1, h_2) \in \mathbb{R}^2_+ : h_1 = h_2 < \frac{10}{\sqrt{21}}\operatorname{acos}\left(-\frac{10}{11}\right) \approx 5.92\right\}.$$

From Figure 3.1 it is evident that such a coarse analysis does not reveal that the size of the stability region along a ray is very sensitive with respect to changes in the fraction $\frac{h_1}{h_2}$, i.e., the slope of the ray in delay-space. The same possibility to miss large parts of the stability region occur for any ray in delay-space, e.g. when the delays are commensurate.

In fact, the situation can be even worse. Some directions in delay-space may be stable independent of the size of the delay (i.e. $\tau = \infty$), whereas arbitrarily small perturbations of the slope of the ray makes τ finite. This is known as the *delay interference phenomenon* [MN07a].

In this type of analysis the result of the analysis is one number, τ. This information is easy to interpret, but it is of course only a sufficient condition for stability. It does not reveal a lot of information about the geometry of D_s.

S2 One definition of the *stability radius* is the largest τ such that the system is stable for all $h_1 < \tau$ and $h_2 < \tau$, i.e., geometrically, the largest square with one corner in the origin completely contained in D_s. In terms of sets, we wish to find the largest τ such that

$$\{(h_1, h_2) \in \mathbb{R}^2_+ : (h_1 < \tau) \wedge (h_2 < \tau)\} \subset D_s.$$

For this problem, the largest τ is given by

$$\tau = h_2 = \frac{10}{\sqrt{399}}\operatorname{acos}\left(-\frac{1}{20}\right) \approx 0.811.$$

It is clear from Figure 3.1(b), that this type of stability analysis gives a very conservative bound in the sense that a lot of information about D_s can not be identified. For instance, this analysis does not reveal that if h_2 is small, the system is stable independent of h_1.

S3 One can think of generalizations of the stability radius in S2. For instance, we can define the radius as the largest pair $\tau = (\tau_1, \tau_2)$ such that the system is stable for all points inside the delay-space rectangle defined by τ_1 and τ_2. Equivalently in terms of sets,

$$\{(h_1, h_2) \in \mathbb{R}^2_+ : (h_1 < \tau_1) \wedge (h_2 < \tau_2)\} \subset D_s.$$

To pose a well defined maximization problem we must define what is meant by largest τ. The are several natural candidates. For instance, we may optimize with respect to the area of the rectangle, i.e., $\tau_1\tau_2$ or pick a norm for $\mathbb{R}^2$.

For this example, if we pick the area of the rectangle as measure, we get $\tau \approx (\infty, 0.8)$ but if we optimize with respect to τ_2 (and then τ_1) we get $\tau \approx (1.79, 0.811)$.

Even though this method to analyze the stability region is less conservative, i.e., provides more information about D_s, in comparison to S2, the wrong choice of optimization measure may yield very conservative results. Moreover, for this example, the large stable region for $h_1 \approx h_2$ will never be identified.

See [GKC03, Theorem 3.9], [IND+03] for some stability results represented by (unbounded) rectangles.

$\mathcal{T}$ (parameterization) Finally, we have the type of analysis to which the main result of this chapter belongs. We saw that any result yielding a stability statement of the form of S1, S2 or S3 will fail to describe the whole stability region D_s. Since the stability region D_s and the boundary ∂D_s are complicated geometrical objects (D_s is unbounded and ∂D_s not smooth), we instead try an exact description of $\mathcal{T}$, hoping that it is easier to represent. We wish to state such a result by a finite number of explicit formulas containing elementary functions or a finite number of numerically elementary operations. This is the definition of Problem 3.2.

The method for Problem 3.2 which will be presented in Section 3.3 is a parameterization of the set $\mathcal{T}$. Here, (using Corollary 3.17) the parameterization is given by

$$\mathcal{T} = \bigcup_{\substack{\varphi\in[-\pi,\pi]\\ p_1,p_2\in\mathbb{Z}}} (h_1(\varphi), h_2(\varphi)),$$

where

$$\begin{aligned} h_1(\varphi) &= \frac{\varphi + 2q\pi}{\omega(\varphi)}, \\ h_2(\varphi) &= \frac{\operatorname{atan}\left(\begin{smallmatrix}\omega(\varphi)+\frac{9}{10}\sin(\varphi)\\ -1+\frac{9}{10}\cos(\varphi)\end{smallmatrix}\right) + 2p\pi}{\omega(\varphi)}, \\ \omega(\varphi) &= \pm\sqrt{4 - \left(-1 + \frac{9}{10}\cos(\varphi)\right)^2} - \frac{9}{10}\sin(\varphi). \end{aligned}$$

We have denoted the four-quadrant inverse of tangent by $\operatorname{atan}\left(\begin{smallmatrix}a\\ b\end{smallmatrix}\right)$, i.e., $\operatorname{atan}\left(\begin{smallmatrix}a\\ b\end{smallmatrix}\right) = \operatorname{Arg}\,(b + ai)$.

In comparison to the information provided in S1,S2 and S3, the suggested parameterization of $\mathcal{T}$ is somewhat implicit as it is not easy to directly read off as many properties of D_s, ∂D_s or $\mathcal{T}$. It is however an exact description of $\mathcal{T}$. Moreover, since it only consists of elementary functions, theoretical geometric information can be extracted by hand.

Apart from the parameterization presented in this chapter, a parameterization of $\mathcal{T}$ for (some) two-delay DDEs is given in [GNC05]. This work is further discussed in Example 3.11.

3.2 Notes and references on delay-dependent stability results

The delay-dependent stability problem, i.e., Problem 3.4, has received a lot of attention in the last couple of decades.

In order to restrict the discussion to works related to our results, we make the following coarse classification of delay-dependent stability results. Apart from the methods based on computing the spectra, i.e., those reviewed in Chapter 2,

most of the literature on delay-dependent stability-conditions of DDEs can be classified by their main ideas as follows:

- Stability can be stated as the existence of a certain functional, called *Lyapunov-Krasovskii functional* (LKFs). In practice the existence condition of the LKF is typically restated as *linear matrix inequality* (LMI) which is a fairly common problem in the field of convex optimization.
- At a point of where stability switches there is a purely imaginary eigenvalue which can be computed or characterized, i.e., $\partial D_s \subset \mathcal{T}$ (some neutral DDEs excluded).

As mentioned earlier, in this chapter we focus on the second approach. We wish to point out that both types of approaches have advantages and disadvantages. For instance, the methods in [Fri06], [Nic01b] and the methods summarized in [GKC03] are examples of results where a Lyapunov-Krasovskii functional is used to construct sufficient but conservative delay-dependent stability conditions formulated as *linear matrix inequalities* (LMIs). An advantage of these types of approaches is that the conditions can be elegantly formulated with LMIs which allow the automatic application in engineering software. A disadvantage is that the results are often conservative and may impose unnecessarily pessimistic constraints, cf. the stability analysis S2, S3 in Example 3.6. Moreover, the treatment of LMIs of large dimension is currently computationally cumbersome.

In the sequel we will restrict the discussion to methods based on the analysis of imaginary eigenvalues.

The large number of publications with sometimes large intersections make a complete overview difficult. The book by Bellman and Cooke [BC63] still serves as a standard reference in the field of time-delay systems. In order to limit the discussion we will focus on works after the appearance of this book. The book contains a fair amount of references to earlier works on time-delay systems in general and stability conditions based on imaginary eigenvalues such as important earlier methods, e.g. the *Nyquist criterion* and the *root-locus methods*. For more complete overviews of more recent stability conditions and methods for stability analysis, we refer the reader to the book of Niculescu [Nic01a] and the recent book of Michiels and Niculescu [MN07b, Chapter 3-4].

It is not astonishing that works prior to the book of Bellman and Cook have a focus on exact explicit conditions for systems of low order, and that research

after the book has been in the direction of more general systems with expressions computable on a computer.

We mention two representative early results.

Theorem 3.7 ([Hay50] see also [BC63, Theorem 13.8]) *All the roots of* $pe^s + q - se^s = 0$*, where* $p, q \in \mathbb{R}$*, have negative real parts if and only if*

(a) $p < 1$*, and*

(b) $p < -q < \sqrt{a_1^2 + p^2}$

where a_1 *is the root of* $a = p\tan(a)$ *such that* $0 < a < \pi$*. If* $p = 0$*, we let* $a_1 = \pi/2$*.*

Theorem 3.8 ([BC63, Theorem 13.12], see also [Min48]) *Let* $H(s) = s^2 e^s + as + b$ *where* $a, b \in \mathbb{R}$*. Denote by* c_a *the root of the equation (there is such a root if (a) below holds)*

$$\sin(c_a) = a/c_a$$

which lies on the interval $(0, \pi/2)$*. A necessary and sufficient condition that all the roots of* $H(s) = 0$ *lie in* $\mathbb{C}_-$ *is that*

(a) $0 < a < \pi/2$

(b) $0 < b < c_a^2 \cos(c_a)$*.*

In Table 3.1 we have listed some of the important works in the field after 1963. From the table, it is clear that the main focus of the research has shifted. Early research was done on single-delay scalar systems and typically yielding Routh-Hurwitz criteria and exact trigonometric expressions. The recent advances involve state-space representations and exact conditions expressed in terms of eigenvalue problems (matrix pencil methods). In the last couple of years the focus has been to generalize the theory from retarded systems with a single-delay or commensurate systems to neutral systems with multiple delays.

We describe some of the works from Table 3.1 in more detail. Note that in order to keep the discussion consistent and limited we explain some results in a slightly less general setting than in the original works. Moreover, Table 3.1 should be interpreted with care as the characteristic equation does not reveal other results in the analysis. For instance, the authors of [FR86] and [RV07] give a thorough analysis of some classes of *nonlinear* time-delay systems.

Year	Characteristic equation	Comment	References
1964	$s^3 + a_1 s^2 + a_2 s + a_3 + a_4 e^{-\tau s} = 0$	Classification	[Muf64]
1969	$P(s) + Q(s)e^{-\tau s} = 0$, retarded	τ-decomposition	[LH69]
1980	$-s + \sum_{k=0}^{m} a_k(s) e^{-skh} = 0$	Rekasius	[Rek80] [Tho81] [HJZ84]
1982	$P(s) + Q(s)e^{-\tau s} = 0$	Geometric	[Mah82]
1982	$s^2 + as + bse^{-\tau s} + c + de^{-\tau s} = 0$	Trigonometric	[CG82]
1986	$P(s) + Q(s)e^{-\tau s} = 0$	Trigonometric	[CvdD86] [Boe98]
1986	$\det\left(-sI + \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix} + \begin{pmatrix} b_0 e^{-s\tau_1} & c_1 e^{-s\tau_2} \\ b_2 e^{-s\tau_1} & c_3 e^{-s\tau_2} \end{pmatrix}\right) = 0$	Estimates	[FR86]
1987	$P(s) + Q(s)e^{-\tau s} = 0$	Polynomial	[WM87] [BP07]
1989	e.g. $s^2 + a + be^{-\tau_1 s} + be^{-\tau_2 s} = 0$	Trigonometric	[Sté89]
1991	$s + a + be^{-s\tau_1} + ce^{-s\tau_2} = 0$	Classification	[Hal91],[HH93]
1994	$-s + e^{-s\tau_1} + ae^{-s\tau_2} = 0$	Geometric	[BC94], [LRW99], [Pio07]
1995	$s^2 + s\sum a_k e^{-s\tau k} + \sum b_k e^{-s\tau k} = 0$	Geometric	[Boe95]
1995	$\det\left(-sI + \sum_{k=0}^{m} A_k e^{-k\tau s}\right) = 0$	Matrix pencil	[CGN95], [Nic98],[NFC05]
1995	$s + a + be^{-\tau_1 s} + ce^{-\tau_2 s} = 0$	Trigonometric	[MJZ95]
1996	$\det\left(-sI + A_0 e^{-hs}\right) = 0,\ A_0 \in \mathbb{R}^{2\times 2}$	Trace, Norm	[HS96]
2000	$\det(-sI + \alpha A + (1-\alpha)Ae^{-\tau s}) = 0$	Geometric	[CS00]
2001	$\det\left(s(I + B_1 e^{-hs}) - (A_0 + A_1 e^{-hs})\right) = 0$	Matrix pencil	[Lou01]
2002	$P(s,\tau) + Q(s,\tau)e^{-\tau s} = 0$	Trigonometric	[BK02]
2002	$\det\left(sI + A_0 + A_1 e^{-\tau s}\right) = 0 = \sum_{k=0}^{n} a_k(s) e^{-k\tau s} = 0$	Routh table	[OS02]
2004	$P(s,\tau) + Q(s,\tau)e^{-s\tau} + R(s,\tau)e^{-2s\tau} = 0$	Trigonometric	[LM04]
2004	$a_0 + a_1 e^{-\tau s} + b_0 s + b_1 s e^{-\tau s} + s^2 = 0$	Trigonometric	[CS04]
2004	$\det\left(-sI + A_0 + A_1 e^{-\tau s}\right) = 0$	Routh table	[OS04]
2005	$p_0(s) + p_1(s)e^{-\tau_1 s} + p_2(s)e^{-\tau_2 s} = 0$	Parameterization	[GNC05]
2005	$\det\left(-sI + A_0 + A_1 e^{-\tau_1 s} + A_2 e^{-\tau_2 s}\right) = 0,\ n = 3$	Routh array	[SO05]
2005	$s^4 + as^3 + bs^2 + cs + d + re^{-s\tau} = 0$	Trigonometric	[LW05]
2006	$s(1 - ae^{-\tau_1 s} - be^{-\tau_2 s})$ $-ce^{-\tau_1 s} - de^{-\tau_2 s} - fe^{-(\tau_1+\tau_2)s} - g = 0$	Routh table	[SO06a]
2006	$s^2 + a_1 s + a_0 + (b_2 s^2 + b_1 s + b_0)e^{-sh} = 0$	Trigonometric	[MM06]
2006	$\det(-sI + A_0 + A_1 e^{-\tau s}) = 0$	Single delay	[SO06b]
2006	$\det\left(-B_0 s + \sum_{k=0}^{m} A_k e^{-k\tau s}\right) = 0,\ \det(B_0) = 0$	Matrix pencil	[NFC06]
2006	$\det\left(-sI + \sum_{k=0}^{m} A_k e^{-\tau_k s}\right) = 0$	Matrix pencil	[Jar06a] (Section 3.3)
2006	$\det\left(-s(I + \sum_{k=1}^{m} B_k e^{-k\tau s}) + \sum_{k=0}^{m} A_k e^{-k\tau s}\right) = 0$	Matrix pencil	[FNC06]
2006	$s + ae^{-\tau_1 s} + be^{-\tau_2 s} = 0$	Trigonometric	[Lev06]
2007	$\det\left(-s\left(\sum_{k=0}^{m} B_k e^{-\tau_k s}\right) + \sum_{k=0}^{m} A_k e^{-\tau_k s}\right) = 0$	Matrix pencil	[Jar07] (Section 3.4)
2007	$-s + a + b^{-s\tau} = 0$	Nonlinear	[RV07]
2007	$a(\tau)s + b(\tau) + c(\tau)e^{-s\tau} + d(\tau)e^{-2s\tau} = 0$	Trigonometric	[SLHT07]
2007	$\det\left(-sI + \sum_{k=0}^{m} A_k e^{-\tau_k s}\right) = 0$	Sum of sq.	[MEA07]
2007	$\det\left(-s(I + \sum_{k=1}^{m} B_k e^{-k\tau s}) + \sum_{k=0}^{m} A_k e^{-k\tau s}\right) = 0$	Multivariate	[MVZ+07]
2007	$\det\left(-sI + \sum_{k=0}^{m} e^{-\tau_k s}\right) = 0$	Matrix pencil	[EOF07]
2008	$\det(s(I - Ae^{-\tau_1 s} - Be^{-\tau_2 s} - Ce^{-(\tau_1+\tau_2)s})$ $- De^{-\tau_1 s} - Fe^{-\tau_2 s} - Ge^{-(\tau_1+\tau_2)s} + H) = 0$	Routh, QPMR	[OVS08]

Table 3.1: Some stability conditions based on imaginary eigenvalues between 1963-2007, i.e., from the book of Bellman and Cooke [BC63] until present.

Geometric/trigonometric analysis

The paper by Cooke and Grossmann [CG82] is a commonly cited important work on exact stability conditions for single-delay scalar first-order and some second-order systems. This is a clear example how explicit exact stability conditions can sometimes be expressed as trigonometric functions. The most general result is

for equations of the type $\ddot{x}(t) + a\dot{x}(t) + b\dot{x}(t-\tau) + cx(t) + dx(t-\tau) = 0$. The only possible imaginary eigenvalues for this DDE are $i\omega$, where

$$\omega = \pm\sqrt{b^2 - a^2 + 2c \pm \sqrt{\frac{1}{4}(b^2 - a^2 + 2c)^2 - (c^2 - d^2)}}.$$

The critical delays are

$$\tau = \frac{1}{\omega}\left(\operatorname{acos}\left(\frac{d(\omega^2 - c) - \omega^2 ab}{b^2\omega^2 + d^2}\right) + 2p\pi\right).$$

Some parts of this work were generalized in [CvdD86] and later corrected in [Boe98]. It was shown that the critical delays are characterized by the two equations

$$\begin{aligned}\sin(\tau\omega) &= \frac{P_R(\omega)Q_I(\omega) + Q_R(\omega)P_I(\omega)}{Q_R^2 + Q_I^2} \text{ and}\\ \cos(\tau\omega) &= \frac{-(P_R(\omega)Q_R(\omega) + Q_I(\omega)P_I(\omega))}{Q_R^2 + Q_I^2},\end{aligned}$$

where the P_R, P_I, Q_R and Q_I are the real/imaginary parts of the linear and exponential polynomials in the characteristic equation $p(i\omega) = P(i\omega) + Q(i\omega)e^{-i\omega\tau}$. Furthermore, the crossing direction, i.e., the sign of the derivative with respect to the delay at the critical imaginary eigenvalue $i\omega$ is given by

$$\text{sign Re}\,\frac{ds}{d\tau} = \text{sign}\,(P_R(\omega)P_R'(\omega) + P_I(\omega)P_I'(\omega) - Q_R(\omega)Q_R'(\omega) - Q_I(\omega)Q_I'(\omega)).$$

Another related geometric analysis is performed in [BK02] for the case that the coefficients are dependent on the delays. The setting of the work is a general single-delay setting, with characteristic equation $p(\lambda, \tau) = P(\lambda, \tau) + Q(\lambda, \tau)e^{-\lambda\tau}$ for given functions P and Q which are polynomials in λ. The authors present necessary conditions for a stability switch. The condition is expressed using the roots of the nonlinear equation $F(\omega, \tau) = |P(i\omega, \tau)|^2 - |Q(i\omega, \tau)|^2$. The coefficients are dependent on the delay in some applications from population dynamics.

The retarded scalar two-delay systems were further analyzed in [Hal91] and [HH93], where the shape of the stability region as well as the *Hopf bifurcation curves* (critical curves) were classified using a geometric interpretation of constructed comparison functions. In a sense, this work is extended in [GNC05]

by generalizing it to some neutral systems and giving a parameterization of the critical delays consisting of trigonometric expressions. We give an explicit example of that method in Example 3.11. The analysis is relevant in applications in population dynamics models for leukemia. [NKGL06] .

Exact conditions for second order systems with a single delay are revisited and reformulated in [MM06].

We now mention some less known works where the stability of some DDEs can be expressed as inequalities in terms of trace and norm. Explicit conditions for time-delay systems containing only delay and differential-terms, i.e., $\dot{x}(t) = Ax(t-\tau)$ where $A \in \mathbb{R}^{2\times 2}$ are treated in [HS96]. The conditions are compactly stated in terms of trace of A and matrix-norms. One of the results is: The system is stable if and only if

$$2\sqrt{||A||}\sin(\tau\sqrt{||A||}) < -\mathrm{Tr}A < \frac{\pi}{2\tau} + \frac{2\tau||A||}{\pi} \tag{3.9}$$

and

$$0 < \tau^2||A|| < \left(\frac{\pi}{2}\right).$$

See also the results and reviews in [Mat07].

Rekasius-type analysis

Methods based on the *Rekasius substitution* represent another set of methods used to analyze delay-space stability. The method can be used to find conditions such that the delay-system has a purely imaginary eigenvalue by transforming the characteristic equation containing exponential terms to a polynomial. In its simplest form the Rekasius substitution does the following: The problem of determining $h \in \mathbb{R}^+$ such that $-i\omega + a + be^{-hi\omega} = 0$ is replaced by the problem to find $T \in \mathbb{R}$ (sometimes called pseudodelay) such that

$$-i\omega + a + b\frac{1 - i\omega T}{1 + i\omega T} = 0.$$

This equation can be solved for T and ω by separating real and imaginary parts, which both are polynomials. Once T and ω are found, h can be computed by inverting the transformation $e^{-hi\omega} = \frac{1-i\omega T}{1+i\omega T}$, i.e., $h = \frac{2}{\omega}[\mathrm{atan}(\omega T) + p\pi]$ for any $p \in \mathbb{Z}$. We note that the Rekasius substitution is in other contexts called

a *Möbius transformation* (bilinear transformation) and is a special case of the *Cayley transformation* of e^{-hs}.

These ideas were presented in [Rek80] and [Tho81]. It was further developed in [HJZ84] (and references therein) where some inconsistencies were corrected by introducing an additional Routh-Hurwitz type test. A related method (called the *direct method*) was presented in [WM87] (reinterpreted in [Par04a]) and generalized to fractional delay systems in [BP07] where the idea is to find s and h such that $p(s,h) = 0$ and $p(-s,h) = 0$ for the characteristic quasi-polynomial p. A Rekasius approach was also recently used in a series of papers on time-delay systems with multiple delays ([OS02], [OS03], [SO05], [SO06c], [FSO07]) and on a restricted set of neutral systems in [OS04] and [SO06a]. For the multiple-delay case the method allows the computation of the points where the eigenvalues cross the imaginary axis (called crossing frequency) by setting up a parameterized Routh array. These works also contain an analysis of the root-crossing direction (called the root-invariance property) which allows the computation of the number of unstable roots for each stability region in delay-space in a semi-automatic manner.

The Rekasius substitution was also used in a work by Ebenbauer and Allgöwer [EA06] and Münz et al. [MEA07]. The authors derive sufficient conditions by combining the Rekasius substitution with the sum of squares technique.

Matrix pencil methods

For larger problems, most of the results mentioned above are unsuitable as they are based on a reasoning with scalar conditions. Some more recent results are stated directly in terms of the system matrices without computing the coefficients in the characteristic equation. One such group of results are those based on eigenvalues of a matrix pencil. The simplest form is the characterization of the critical delays for a DDE with a single delay $\dot{x}(t) = A_0 x(t) + A_1 x(t-h)$. They can be computed from the expression

$$h = \frac{\operatorname{Arg} z + 2p\pi}{\omega}, \text{ where } z : \det(z^2 I \otimes A_1 + z(I \otimes A_0 + A_0 \otimes I) + A_1 \otimes I) = 0, \\ \text{and } i\omega \in \sigma(A_0 + A_1 z),$$

where $\otimes$ is the Kronecker product as usual. This can be derived from Theorem 3.18. A similar expression can be found for DDEs with commensurate

delays in [CGN95] and for neutral DDEs in [Lou01].

To the authors knowledge, the following result was the first matrix pencil method for neutral DDEs.

Theorem 3.9 ([Lou01, Theorem 3.1]) *Let $A_0, A_1, B_1 \in \mathbb{R}^{n\times n}$. Then all imaginary axis eigenvalues of the DDE*

$$\dot{x}(t) + B_1\dot{x}(t-\tau) = A_0x(t) + A_1x(t-\tau) \tag{3.10}$$

are zeros of $\det((sI - A_0)\otimes(sI + A_0) - (sB_1 - A_1)\otimes(sB_1 + A_1)) = 0$ *and thus also of*

$$\det\left(s\begin{pmatrix} I\otimes I & B_1\otimes I \\ I\otimes B_1 & I\otimes I\end{pmatrix} - \begin{pmatrix} A_0\otimes I & A_1\otimes I \\ -I\otimes A_1 & -I\otimes A_0\end{pmatrix}\right) = 0. \tag{3.11}$$

If $s \in i\mathbb{R}$ is an eigenvalue of (3.10) *with associated eigenvector v then $w =$ vec vv^* is the kernel corresponding to the matrix pencil in* (3.11).

The first matrix pencil for commensurate DDEs was given in [CGN95] and extended with a formula for the switching direction in [NFC05]. These results were generalized to delay-differential algebraic equations (DDAEs) in [NFC06] and to neutral systems in [FNC06].

One way to describe $\mathcal{T}$ for DDEs with multiple delays, is (as mentioned) to give a parameterization of $\mathcal{T}$. The parameterizations presented in Section 3.3 for retarded DDEs and for neutral DDEs in Section 3.4, are also matrix pencil methods because each evaluation involves the solution of a quadratic eigenvalue problem. Finally, the method in [EOF07] is also a matrix pencil method for DDEs with multiple delays.

It is worthwhile to discuss technical and presentational details of [EOF07] since the parameterization presented in Section 3.3 and the method presented in [EOF07] are equivalent in exact arithmetic. Parts of the results of Section 3.3 were published in proceedings of international conferences [Jar06a]. Early results of [EOF07] were presented in the proceedings of the same international conference [EO06].

In the derivation in Section 3.3 we construct a *matrix equation* which can be vectorized into a quadratic eigenvalue problem with matrix-coefficients expressed with Kronecker products. The derivation in [EOF07] is done entirely in vectorized

form, i.e., the authors construct the sum of Kronecker products (Kronecker sums) with the characteristic equation.

The focus in Section 3.3 is to construct a parameterization, i.e., a mapping containing computable expressions, and not to efficiently numerically compute the critical delays . The method in [EOF07] is presented with the idea in mind that a numerical procedure should be produced. The method constructed is shown to be more efficient than some numerical procedures in previous publications of the authors. It is easy to think of other efficient numerical schemes, e.g. the natural ideas to numerically compute $\mathcal{T}$ in Remark 3.3.

In the numerical examples, the authors of [EOF07] suggest to explicitly compute the coefficients in the characteristic polynomial (of the quadratic eigenvalue problem). This is known to be a numerically unstable representation of an eigenvalue problem, and particularly unsuitable for a numerical procedure.

Furthermore, even though the numerical procedure "works" for many practical cases, it easy to think of technical ambiguities for the definitions and theorems of [EOF07]. For instance, the case that the origin can be an eigenvalue is not properly excluded. Trivially, if all matrices are zero, then $s = 0$ is the only eigenvalue for any choice of the delays. Since several expressions contain $1/\omega$ the results are not well defined. Even in the limit, there are ambiguities. The (so-called) kernel hypersurfaces in [EOF07, Definition 1] contain $1/\omega$, which in the limit means that the *kernel hypersurfaces* are in the limit (roughly speaking) $\mathbb{R}^m_+$. In [EOF07, Corollary 1] it is stated that the number of kernel hypersurfaces is bounded by n^2. It is not clear how *number of kernel hypersurfaces* is defined. In particular, there is no obvious definition if $\omega = 0$ or in the limit case. The same ambiguities occur for other singular DDEs.

Finally, in [EOF07], the numerical difficulties of the methods are discussed and related to the NP-hardness results of Toker and Özbai [TÖ96]. The results of Toker and Özbai have no impact on a numerical procedure for the critical delays. See the discussion in Section 3.7.

3.3 A parameterization for retarded DDEs

The main problem of this chapter, i.e., Problem 3.2, is to determine an exact representation of $\mathcal{T}$ which is useful in the sense that much stability information

is available by inspection or by using a finite number of elementary operation.

The representation we suggest is a parameterization. In our context, a parameterization is a mapping from a simple mathematical object, which we will (for the moment) call X, such that the union of the range of the mapping is $\mathcal{T}$. That is, $h : X \to \{\mathbb{R}^m_+\}$ is a parameterization of $\mathcal{T}$ if it has the surjective property,

$$\mathcal{T} = \bigcup_{x \in X} h(x). \tag{3.12}$$

Note that h is a mapping from X onto the set of all subsets of $\mathcal{T}$. Clearly, if $h(x)$ consists of only one element for all $x \in X$, then (3.12) is the definition of a surjective mapping onto $\mathcal{T}$.

In order to have a result useful in practice, we want a mapping which is explicit and consists of a finite number of elementary operations. At least, it should be possible to evaluate h (at least on a computer) with moderate computational effort.

In the parameterization we present (essentially given in Theorem 3.14), $X = [-\pi, \pi]^{m-1} \times \mathbb{Z}^m$ and the mapping h is expressed in terms of trigonometric expression for the scalar case, i.e., $n = 1$, and consists of a *quadratic eigenvalue problem* for the general case $n > 1$. Quadratic eigenvalue problems can be rewritten into a linear eigenvalue problem (see Appendix A.1), i.e., the problem of determining the eigenvalues of a matrix. The eigenvalue problem is one of the fundamental problems in numerical linear algebra and can be solved to sufficient precision in a reasonable amount of time if the dimension is moderate. In our case, $h(x)$ is a finite set if the quadratic eigenvalue problem does not have infinite eigenvalues (since a quadratic eigenvalue problem of dimension N can have at most $2N$ eigenvalues).

In order to make the presentation smoother in the example that follows, we sketch the derivation for the scalar DDE with two delays and make notes how the result changes for the general case.

Example 3.10 *Suppose we have a purely imaginary eigenvalue $s = i\omega$ with corresponding eigenvector v, i.e., $M(i\omega)v = 0$. The characteristic equation for a DDE with two delays is*

$$M(i\omega)v = \left(-i\omega I + A_0 + A_1 e^{-ih_1\omega} + A_2 e^{-ih_2\omega}\right) v = 0.$$

We now set $\varphi := h_1\omega$ *as a free parameter in the parameterization and denote* $z := e^{ih_2\omega} \in \partial\mathbb{D}$. *As usual we denote the unit disc by* $\mathbb{D} := \{z \in \mathbb{R} : |z| < 1\}$ *and* $\partial\mathbb{D}$ *the unit circle. The characteristic equation is now*

$$M(i\omega)v = \left(-i\omega I + A_0 + A_1 e^{-i\varphi} + A_2 z\right) v = 0,\ v \neq 0. \tag{3.13}$$

Clearly, it is 2π*-periodic in* φ, *it will hence be enough to let* φ *run through the whole span* $[-\pi, \pi]$. *That is, for each choice of* φ *in this interval we will find some points on the critical curves. We will see later that it does indeed have the surjective property* (3.12).

If we first consider the scalar case, i.e., $A_0 = a_0, A_1 = a_1, A_2 = a_2 \in \mathbb{R}$ *the equation* (3.13) *corresponds to two scalar conditions (say real and imaginary parts) and we can eliminate either* ω *or* z. *In the approach presented here, we eliminate* ω *by forming the sum of* (3.13) *and its complex conjugate, i.e.,*

$$0 = 2a_0 + a_1 e^{-i\varphi} + a_2\bar{z} + a_1 e^{i\varphi} + a_2 z = a_2\bar{z} + 2a_0 + 2a_1\cos(\varphi) + a_2 z. \tag{3.14}$$

Multiplying with z *yields the quadratic equation,*

$$a_2 z^2 + 2z(a_0 + a_1\cos(\varphi)) + a_2 = 0,$$

since $z\bar{z} = 1$. *It has the two solutions*

$$z = \frac{-(a_0 + a_1\cos(\varphi)) \pm i\sqrt{a_2^2 - (a_0 + a_1\cos(\varphi))^2}}{a_2},$$

assuming $a_2 \neq 0$. *We can now give a formula for* ω *by inserting* z *into* (3.13) *and rearrange the terms, i.e.,*

$$i\omega = a_0 + a_1 e^{-i\varphi} + a_2 z = i\left(-a_1\sin(\varphi) \pm \sqrt{a_2^2 - (a_0 + a_1\cos(\varphi))^2}\right).$$

Since $z = e^{-ih_2\omega}$ *and* $\varphi = h_1\omega$, *a parametrization of the critical curves is*

$$\bar{h}(\varphi) = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = \begin{pmatrix} \frac{\varphi + 2p\pi}{-a_1\sin(\varphi) \pm \sqrt{a_2^2 - (a_0 + a_1\cos(\varphi))^2}} \\ \frac{-\text{Arg } z + 2q\pi}{-a_1\sin(\varphi) \pm \sqrt{a_2^2 - (a_0 + a_1\cos(\varphi))^2}} \end{pmatrix}, \tag{3.15}$$

where $\text{Arg } z = \pm\text{sign } (a_2)\text{acos}\left(-\frac{a_0 + a_1\cos(\varphi)}{a_2}\right)$ *for any* $p, q \in \mathbb{Z}$ *and* $\varphi \in [-\pi, \pi]$. *Note that the signs must be matched, i.e., for each choice of the free parameter* φ *and branches* p *and* q, *the parameterization has two critical delays.*

For the more general case, i.e., an arbitrary number of delays and any dimension of the system the same general elimination can be carried out. The number of delays is compensated by introducing more free parameterization variables. If the system matrices A_0, A_1 and A_2 are not scalar, the elimination of ω in equation (3.14) *must be modified. This step involves taking the sum of* (3.13) *and its complex conjugate. Since v is typically a complex vector, the sum of* (3.13) *and its complex conjugate will not eliminate ω. Taking the sum of* (3.13) *and its complex conjugate transpose is of course also not possible as the dimensions do not fit. Instead we form the sum of $M(s)vv^*$ and $vv^*M(s)^*$, where A^* denotes the complex conjugate transpose of A. We believe this is a natural way to generalize the elimination of ω. Clearly,*

$$\begin{aligned} 0 = M(s)vv^* + vv^*M(s)^* = \\ = A_2vv^*z + ((A_0 + A_1e^{-i\varphi})vv^* + vv^*(A_0^T + A_1^Te^{i\varphi})) + vv^*A_2^T\bar{z}, \end{aligned}$$

which is a matrix equation. This equation can be rewritten into a quadratic eigenvalue problem *by vectorization, i.e., stacking the columns of the matrix on top of each other. Quadratic eigenvalue problems can be rewritten into a generalized eigenvalue problem using a so-called* companion linearization *(see Appendix A.1). As we will see later, if n is not large, there are numerical methods to find all z. We can then compute the corresponding ω and the critical delays in a similar way as for the scalar case.*

Even though the motivation above is simple, it is not clear that all steps involved are reversible, i.e., there is no convincing argument to ensure that we get all critical delays. For that reason, we formalize the discussion in the sections that follow. At the same time we generalize the method to an arbitrary number of delays. The generalization is such that if we have m delays there are $m-1$ free parameters. We need to solve a quadratic eigenvalue problem for each choice of the parameters. In order to compare the method with related approaches we also discuss the case where the delays are (fixed) integer multiples of the first delay, i.e., commensurate delays.

The parameterization of $\mathcal{T}$ we present is not the first in the literature. To the author's knowledge, it is however the first parameterization for an arbitrary number of delays and arbitrary dimension.

The analysis of Gu et al. [GNC05] (see also [Mor06, Chapter 3] and [MN07b, Chapter 3]) can be interpreted as a parameterization of $\mathcal{T}$ for some DDEs with

two delays. They treat $\omega \in \mathbb{R}$, i.e., the imaginary eigenvalue (frequency), as free parameter. This method has been successfully used to describe stability properties of the *Smith predictor* [MNG07]. Similar ideas were used for DDEs with a distributed delay-term which has applications in population dynamics and in particular Cushings linear equation [MNG06] and [MN02].

In our context, they construct a parameterization $h : X \to \mathbb{R}_+^2$ with $X = \mathbb{R}_+ \times \mathbb{Z}^2$. The main result can be stated as follows. If the characteristic equation can be rewritten as

$$1 + a_1(s)e^{-h_1 s} + a_2(s)e^{-h_2 s} = 0, \tag{3.16}$$

where a_1 and a_2 are rational functions, a parameterization of the critical curves is given by

$$\begin{aligned} h_1 &= \frac{\text{Arg }(a_1(i\omega)) + (2p-1)\pi \pm \theta_1}{\omega}, \quad \theta_1 = \text{acos}\left(\frac{1 + |a_1(i\omega)|^2 - |a_2(i\omega)|^2}{2|a_1(i\omega)|}\right), \\ h_2 &= \frac{\text{Arg }(a_2(i\omega)) + (2q-1)\pi \mp \theta_2}{\omega}, \quad \theta_2 = \text{acos}\left(\frac{1 + |a_2(i\omega)|^2 - |a_1(i\omega)|^2}{2|a_2(i\omega)|}\right). \end{aligned}$$

Note that not all characteristic equations can be written on the form (3.16).

Example 3.11 (Different parameterizations) *In order to demonstrate the difference between* (3.15) *and the parameterization in [GNC05], we construct the parameterizations of the critical curves for the case that* $a_0 = a_1 = -1$, $a_2 = -\frac{1}{2}$ *(taken from [MN07a, Figure 5.1]). According to* (3.15)*, the critical curves are given by,*

$$h(\varphi) = \begin{pmatrix} \frac{\varphi + 2p\pi}{\sin(\varphi) \pm \frac{1}{2}\sqrt{5 + 8\cos(\varphi) + 4\cos^2(\varphi)}} \\ \frac{\pm \text{acos}(-2 - 2\cos(\varphi)) + 2q\pi}{\sin(\varphi) \pm \frac{1}{2}\sqrt{5 + 8\cos(\varphi) + 4\cos^2(\varphi)}} \end{pmatrix}.$$

The set critical curves are shown in Figure 3.2. The minimum of the 2-norm of $h(x)$*, is* $\|h(x)\|_2 \approx 2.896$ *and taken at* $x \approx -0.7012$ *where* $\omega \approx 1.1139$, $h_1 \approx 2.1078$ *and* $h_2 \approx 1.9853$.

In the context of [GNC05], $a_1(i\omega) = \frac{1}{1+i\omega}, a_2(i\omega) = \frac{1}{2(1+i\omega)}$ *and the parame-*

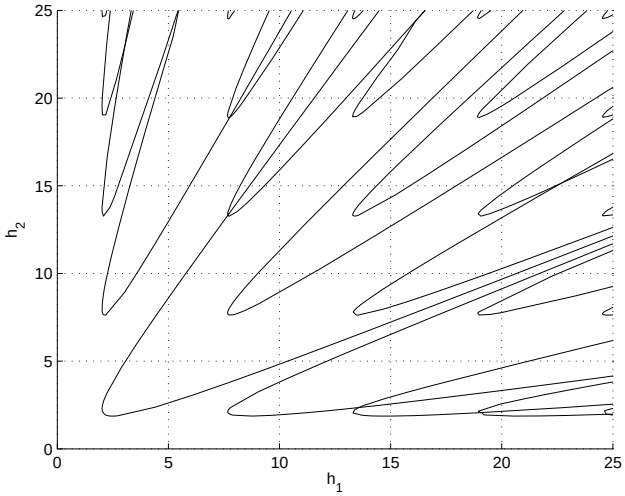


Figure 3.2: The set of critical curves $\mathcal{T}$ for Example 3.11

terization is given by

$$\begin{aligned} h_1 &= \frac{-\text{Arg }(1+i\omega)+(2p-1)\pi \pm \theta_1(\omega)}{\omega} = \\ &= \frac{1}{\omega}\left(-\text{atan}(\omega) \pm \text{acos}\left(\frac{1}{2}\sqrt{1+\omega^2}+\frac{3}{8\sqrt{1+\omega^2}}\right)+(2p-1)\pi\right) \\ h_2 &= \frac{-\text{Arg }(1+i\omega)+(2q-1)\pi \mp \theta_2(\omega)}{\omega} \\ &= \frac{1}{\omega}\left(-\text{atan}(\omega) \mp \text{acos}\left(\sqrt{1+\omega^2}-\frac{3}{4\sqrt{1+\omega^2}}\right)+(2q-1)\pi\right), \end{aligned}$$

which represent the same set of curves, but has a surprisingly small obvious similarity with the other parameterization (3.15)*.*

3.3.1 Main results

We present the parameterization as follows. An equivalence between the characteristic equation and a matrix equation involving the parameter dependent matrix $L(X,s) := M(s)X + XM(s)^*$ is first given in Lemma 3.12. The operator L takes

a particularly simple form for purely imaginary eigenvalues, which makes it possible to state a parameterization containing a matrix equation (Theorem 3.14). It is desirable that the parameterization consists only of formulas containing elementary operations. Here, this is possible because the matrix equation in Theorem 3.14 can be vectorized yielding a quadratic eigenvalue problem. For the scalar case, the quadratic eigenvalue problem is a quadratic equation which we can solve explicitly and express the parameterization in terms of trigonometric expressions.

Lemma 3.12 *Suppose M is a parameter dependent matrix $M : \mathbb{C} \to \mathbb{C}^{n\times n}$. Let $s \in \mathbb{C}$, $v \in \mathbb{C}^n$, $v^*v = 1$ and $L : (\mathbb{C}^{n\times n}, \mathbb{C}) \to \mathbb{C}^{n\times n}$ defined by $L(X, s) := M(s)X + XM(s)^*$, then the following statements are equivalent:*

$$M(s)v = 0 \tag{3.17}$$

$$L(vv^*, s) = 0 \wedge v^*M(s)v = 0. \tag{3.18}$$

Proof: The forward implication is clear from definitions. The backward implication, i.e., (3.18)⇒(3.17), is clear from the identity

$$M(s)v = L(vv^*, s)v - vv^*M(s)^*v.$$

□

In particular, if M is the characteristic matrix function (3.3) then $s = i\omega$ is an eigenvalue if and only if

$$0 = \sum_{k=0}^{m} \left(A_k vv^* e^{-ih_k\omega} + vv^* A_k^T e^{ih_k\omega}\right) \text{ and } \left(i\omega = \sum_{k=0}^{m} v^* A_k v e^{-ih_k\omega}\right).$$

Note that we have eliminated the linear term $-sI$ in M when constructing L. Similar to the sketch in Example 3.10 we introduce the free variables $\varphi_k = h_k\omega$, $k = 1, \ldots, m-1$ and let $z = e^{-ih_m\omega}$. To simplify the formulation of the main theorem the following singular case is treated separately.

Lemma 3.13 *Suppose there is $(h_1, \ldots, h_m) \in \mathbb{R}_+^m$ such that $0 \in \sigma(\Sigma)$, then $\mathcal{T} = \mathbb{R}_+^m$.*

Proof: If $s = 0 \in \sigma(\Sigma)$ for $h_1, \ldots, h_m \in \mathbb{R}_+^m$ and $h_0 = 0$ then

$$0 = \det\left(-sI + \sum_{k=0}^{m} A_k e^{-h_k s}\right) = \det\left(-sI + \sum_{k=0}^{m} A_k e^{-\tau_k s}\right)$$

for any $\tau_1, \ldots, \tau_m$ and $\tau_0 = 0$. Hence, $0 \in i\mathbb{R}$ is in $\sigma(\Sigma)$ and $\sigma(\Sigma) \cap i\mathbb{R} \supset \{0\} \neq \emptyset$ independently of $\tau_1, \ldots, \tau_m$. □

The parameterization is now given by the following theorem.

Theorem 3.14 *Let Σ be the nonsingular DDE (3.1), i.e., $0 \notin \sigma(\Sigma)$, with the fixed coefficients $A_k \in \mathbb{R}^{n\times n}$, $k = 0, \ldots, m$. For any given combination of $\varphi_k \in [-\pi, \pi]$, $k = 1, \ldots, m-1$, $p_k \in \mathbb{Z}$, $k = 1, \ldots, m$, let $z \in \partial\mathbb{D}$ and $v \in \mathbb{C}^n$, $v^*v = 1$ be a solution of the matrix equation*

$$z^2 A_m v v^* + z\left(\sum_{k=0}^{m-1} A_k v v^* e^{-i\varphi_k} + v v^* A_k^T e^{i\varphi_k}\right) + v v^* A_m^T = 0, \tag{3.19}$$

where $\varphi_0 = 0$. Moreover, let

$$\omega = -i v^*\left(A_m z + \sum_{k=0}^{m-1} A_k e^{-i\varphi_k}\right) v, \tag{3.20}$$

and

$$\begin{aligned} h_k &= \frac{\varphi_k + 2p_k\pi}{\omega},\ k = 1, \ldots, m-1, \\ h_m &= \frac{-\text{Arg } z + 2p_m\pi}{\omega}. \end{aligned}$$

Then $\omega \in \mathbb{R}$ and $(h_1, \ldots, h_m) \in \mathcal{T}$. Furthermore, every point in $\mathcal{T}$ corresponds to (at least) one combination of $\varphi_1, \ldots, \varphi_{m-1}$, $p_1, \ldots, p_m$, i.e., the mapping $h = (\varphi_1, \ldots, \varphi_{m-1}, p_1, \ldots p_m) \mapsto \{(h_1, \ldots, h_m)\}$ is a parameterization and

$$\mathcal{T} = \bigcup_{\varphi_k \in [-\pi,\pi], p_k \in \mathbb{Z}} h(\varphi_1, \ldots, \varphi_{m-1}, p_1, \ldots p_m).$$

Proof: We first prove that $\omega \in \mathbb{R}$. Multiplication of (3.19) from the left with v^* and from the right with v and z yields,

$$z v^* A_m v + \left(\sum_{k=0}^{m-1} v^* A_k v e^{-i\varphi_k} + v^* A_k^T v e^{i\varphi_k}\right) + \bar{z} v^* A_m^T v = 0. \tag{3.21}$$

Denote $\alpha = z v^* A_m v + \sum_{k=0}^{m-1} v^* A_k v e^{-i\varphi_k}$. Note that (3.21) implies that $\alpha + \alpha^* = 0$. Hence, Re $\alpha = 0$ and $\omega \in \mathbb{R}$.

The rest of the proof consists of applying Lemma 3.12 in both directions.

We will first show that the construction of $v \in \mathbb{C}^n$, ω, z, $h_1, \ldots, h_m$ yields that $M(i\omega)v = 0$. From Lemma 3.12 it is sufficient to show that $L(vv^*, i\omega) = 0$ and $v^*M(i\omega)v = 0$. The construction of $h_1, \ldots, h_m$ yields that $z = e^{-ih_m\omega}$ and $e^{-i\varphi_k} = e^{-ih_k\omega}$. Let $v \in \mathbb{C}^n$ satisify (3.19) and $\omega \in \mathbb{R}$ satisfy (3.20). Note that (3.19) is $zL(vv^*, i\omega) = 0$ and (3.20) is $iv^*M(i\omega)v = 0$. Since $z \neq 0$, and $\overline{z} = z^{-1}$ we have that $M(i\omega)v = 0$.

It remains to show that h is surjective, i.e., for every point in $(h_1, \ldots, h_m) \in \mathcal{T}$ there is at least one combination of $\varphi_1, \ldots, \varphi_{m-1}$, $p_1, \ldots, p_m$ fullfilling (3.20) and (3.19). Suppose Σ has eigenvalue $i\omega \neq 0$ for some combination of the delays $h_1, \ldots, h_m$. Again, using Lemma 3.12 and the construction of z, φ_k we have (3.19) and (3.20). □

Note that the mapping h is in fact a mapping from $([-\pi, \pi]^{m-1}, \mathbb{Z}^m)$ onto *the set of all subsets of* $\mathcal{T}$ (and not to $\mathcal{T}$). That is, for each choice of $\varphi_1, \ldots, \varphi_{m-1}$, $p_1, \ldots, p_m$ we may get several elements of $\mathcal{T}$ or possibly $\emptyset$.

For the problem we consider, i.e., Problem 3.2, the goal was to construct a parameterization *consisting of computable expressions*. It remains to show that (3.19) and (3.20) are computable expressions. It turns out that the matrix equation (3.19) can be vectorized into a *quadratic eigenvalue problem*, which can be rewritten into a generalized eigenvalue problem, e.g. by *companion linearization* (see Appendix A.1). Hence, Theorem 3.14 corresponds to a computable parameterization if we can solve the corresponding eigenvalue problem efficiently.

The matrix equation (3.19) can be restated into an eigenvalue problem as follows. If we stack the columns of (3.19) on top of each-other, then

$$\left(z^2 I \otimes A_m + z \sum_{k=0}^{m-1} L_k(\varphi_k) + A_m \otimes I \right) u = 0, \tag{3.22}$$

where $u = \bar{v} \otimes v$ and $L_k(\varphi_k) := I \otimes A_k e^{-i\varphi_k} + A_k \otimes I e^{i\varphi_k}$. As usual $\otimes$ denotes the Kronecker product. This operation of rearranging the equations of a matrix equation into a vector equation is called *vectorization*. Problems of the type $(Mz^2 + Gz + K)v = 0$, e.g., (3.22), are known as a *quadratic eigenvalue problems* in the field of numerical linear algebra. Theoretical and numerical properties of the quadratic eigenproblem are reviewed in the survey article [TM01].

The most commonly used methods used to analyze and numerically solve quadratic eigenvalue problems are based on a transformation of the quadratic eigenvalue problem into a generalized eigenvalue problem. In its simplest form, this operation, often called *linearization* or *companion linearization* here yields,

$$\begin{pmatrix} 0 & I \\ A_m \otimes I & \sum_{k=0}^{m-1} L_k(\varphi_k) \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix} = z \begin{pmatrix} I & 0 \\ 0 & -I \otimes A_m \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix}. \tag{3.23}$$

The formulations (3.23) and (3.22) are equivalent except from possibly the multiplicity of infinite eigenvalues. The equivalence essentially follows from the fact that the first block row in (3.23) is the trivial statement $zu = zu$, and the second block row is (3.22).

Historically, companion linearization of quadratic eigenvalue problems has a strong connection with the corresponding operation on systems of ordinary differential equations. It has been used in the field of ordinary differential for a long time and could possibly be called folklore, at least in the field of ordinary differential equations. However, many companion linearizations and generalizations were only recently classified in [MMMM06b]. The type of appropriate companion linearization typically depend on the structure of the matrices. An adequate (structure preserving) linearization preserves properties such as symmetry and eigenvalue pairing. In particular, a structure preserving linearization is often better because a structured matrix can be represented (on a computer) in such a way that rounding errors do not destroy the symmetry of the eigenvalues. Moreover, structured matrices can often be stored (in memory) with less memory resources.

The structure of (3.22) as well as a structured linearization was recently discussed in [FMMS07]. The quadratic eigenvalue problem (3.22) has the symmetry property that there exists a permutation matrix such that if the matrices are conjugated and their order is reversed, we are back at the original eigenvalue problem. This stems from the fact there is a permutation matrix P such that multiplication of this P from both sides reverses order of terms in a Kronecker product. That is, there is a permutation matrix P (independent of B and C) such that $B \otimes C = P(C \otimes B)P$. The class of quadratic eigenvalue problem (3.22) with this symmetry property were named PCP-eigenvalue problem (permutation-conjugation-permutation) in [FMMS07].

The methods based on linearizations are by far the most common in the literature. There are however also other numerical methods to solve quadratic eigen-

value problems which are not based on linearization, e.g., *Jacobi-Davidson* (for polynomial eigenvalue problems) [SBFvdV96] and *second order Arnoldi* [BS05].

With these methods it is possible (unless n is large) to find the eigenpairs of (3.23) to sufficient accuracy on a computer.

Note that it is not possible (in general) to exactly compute the eigenpairs of an eigenvalue problem (e.g. on a computer) without rounding errors. Hence, a numerically computed eigenvector of a solution to (3.23) will normally not fulfill $u = \bar{v} \otimes v$ (exactly). Since we wish to be able to evaluate the parameterization h to a high accuracy on a computer (only using elementary operations) it remains to show how v can be determined to sufficient accuracy given a numerically approximation of u (computed to sufficient accuracy). But first, we show that $\tilde{v}$ is a good approximation of v, if $\tilde{u} \approx u = \bar{v} \otimes v$ and $\bar{\tilde{v}} \otimes \tilde{v} \approx \tilde{u}$.

Lemma 3.15 *Let (z, u) be an eigenpair of the quadratic eigenvalue problem $(z^2M+zG+K)u = 0$ and $u = \bar{v}\otimes v$, $\|v\| = 1$, $|z| = 1$. Say $\tilde{u} = u+\delta_1 = \bar{\tilde{v}}\otimes\tilde{v}+\delta_2$ then the sine of the angle between the approximation v and the vector $\tilde{v}$ is bounded by*

$$|\sin(v, \tilde{v})| \leq \sqrt{\|\delta_1\| \left(\|M + I\| + \|G\| + \|K\|\right) + \|\delta_2\|}.$$

Proof: Multiplying the quadratic eigenvalue problem from the left with $\tilde{u}^*$ and adding $z^2\tilde{u}^*u$ to both sides, yields

$$\tilde{u}^*uz^2 = \tilde{u}^*(z^2(M + I) + zG + K)u.$$

We exploit that $\tilde{u}^*u = (\tilde{v} \otimes \bar{\tilde{v}} + \delta_2)^*(u \otimes \bar{u}) = (\tilde{v}^*v)(\bar{\tilde{v}}^*\bar{v}) + \delta_2^*v$ from which we deduce that

$$\begin{aligned} z^2|\tilde{v}^*v|^2 = z^2(\tilde{v}^*v\bar{\tilde{v}}^*\bar{v}) = \tilde{u}^*(z^2(M + I) + zG + K)u - z^2\delta_2^*u = \\ = u^*(z^2(M + I) + zG + K)u + \delta_2^*(z^2(M + I) + zG + K)u - z^2\delta_2^*x = \\ = z^2 + \delta_1^*(z^2(M + I) + zG + K)u - z^2\delta_2^*u, \end{aligned}$$

i.e.,

$$|\tilde{v}^*v|^2 = 1 + \delta_1^*((M + I) + z^{-1}G + Kz^{-2})u - \delta_2^*u$$

and

$$|\tilde{v}^*v|^2 \geq 1 - \|\delta_1\| \left(\|M + I\| + \|G\| + \|K\|\right) - \|\delta_2\|,$$

where we used that $|z| = 1$. Finally, the proof is completed by the fact that $|\sin(v, \tilde{v})| = \sqrt{1 - |v^*\tilde{v}|^2}$. □

The lemma essentially states that if $\tilde{u}$ is a good approximation of a solution $u = \bar{v} \otimes v$ to the quadratic eigenvalue problem (3.19), i.e., $\|\delta_1\|$ is small, and there is a $\tilde{v}$ such that $\tilde{v} \otimes \bar{\tilde{v}}$ is a good approximation of $\tilde{u}$, i.e., $\|\delta_2\|$ is small, then $\tilde{v}$ is a good approximation of v, i.e., $|\sin(\tilde{v}, v)|$ is small.

Hence, in order to numerically find a solution $\tilde{v}, \tilde{z}$ to (3.19), we should use the approximate solution $\tilde{u}$ of the quadratic eigenvalue problem (3.22), and we need to determine $\tilde{u} = \tilde{v} \otimes \bar{\tilde{v}} + \delta_2$ such that $\|\delta_2\|$ is small.

The problem of determining $\tilde{v}$ given $\tilde{u}$, $\tilde{u} = \tilde{v} \otimes \bar{\tilde{v}} + \delta_2$ such that $\|\delta_2\|$ is small, is a rank-one approximation problem in the following sense. Consider the matrix-version of the conditions, i.e., we wish to find $\tilde{v}$ given $\tilde{U} \in \mathbb{C}^{n \times n}$ (where $\operatorname{vec} \tilde{U} = u$) such

$$\tilde{U} = \tilde{v}\tilde{v}^* + \Delta_2$$

and $\|\Delta_2\|$ is small. That is, we wish to find the best rank-one approximation of $\tilde{U}$. This problem can be numerically solved with the so-called *singular value decomposition* (SVD) . A fundamental property of the singular value decomposition of a matrix M is that the outer product of the left and right vectors corresponding to the largest singular value form the best rank-one approximation of M. Since there are numerical methods for the singular value decomposition, we can numerically compute $\tilde{v}$ from the vector corresponding to the largest singular value.

An important theoretical difference between the standard eigenvalue problem and the generalized eigenvalue problem, e.g. the companion form (3.23), is that generalized eigenvalue problems can have infinite eigenvalues if the right hand side is singular, here if A_m is singular. The following example demonstrates the implications for Theorem 3.14 an infinite eigenvalue can have.

Example 3.16 (Singular case) *Consider the DDE with*

$$A_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, A_1 = \begin{pmatrix} 2 & \varepsilon \\ 3 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

With this example we wish to show how an infinite eigenvalue of (3.22) *can be interpreted, and how this appears in Theorem 3.14. For this particular example it is actually possible to circumvent the problem with the infinite eigenvalue simply by switching A_1 and A_2. Here, we will not approach the problem that way, because that type of reordering of the matrices is not always possible.*

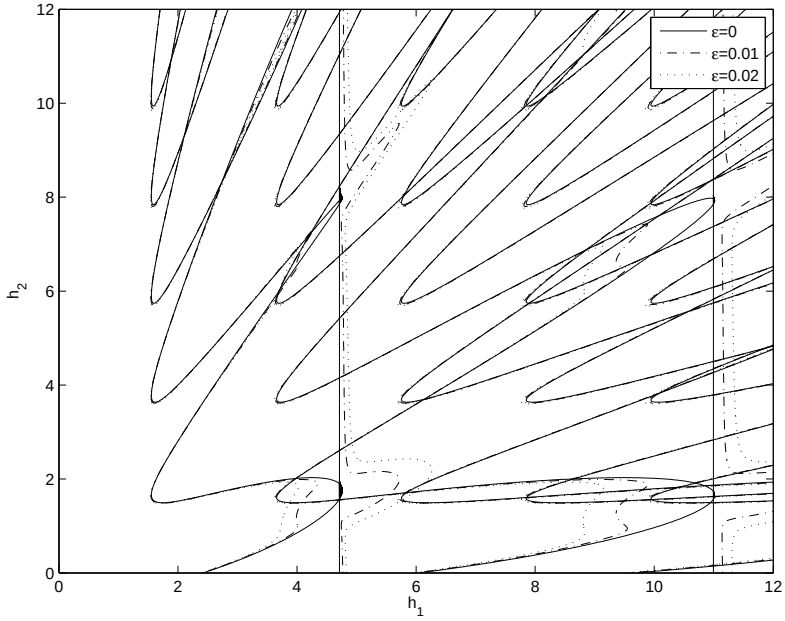


Figure 3.3: Critical curves for Example 3.16

If (3.22) *has an infinite eigenvalue, it must have a corresponding eigenvector such that the quadratic term disappears, i.e., here* $u = (0,1)^T \otimes (0,1)^T = (0,0,0,1)^T$ *is the only eigenvector of the correct form. For this* $v = (0,1)^T$*, the quadratic and the constant term in* (3.19) *are zero, and* (3.20) *is independent of* z*. Since* $z \neq 0$*,* v *is a valid solution of* (3.19) *only if*

$$A_1 v v^* e^{-i\varphi_1} + v v^* A_1^T e^{i\varphi_1} = 0. \tag{3.24}$$

This means that the only choice of φ_1 *which generates a critical delay (corresponding to this* v*) is when* (3.24) *is fulfilled. Here* (3.24) *and hence* (3.19) *is fulfilled only if* $\varepsilon = 0$ *and for* $\varphi_1 = -\frac{\pi}{2} + 2p\pi$ *with the corresponding critical curve* $h_1 = -\frac{\pi}{2} + 2p\pi$ *(for any* h_2*). The critical curves are plotted for three choices of* ε *in Figure 3.3. From the figure it is easy to identify that for* $\varepsilon = 0$ *there are a vertical critical curves, i.e., curves which are independent of* h_2 *corresponding to the contribution of the infinite eigenvalue. If* $\varepsilon > 0$ *there are no vertical lines, no infinite eigenvalues and the critical curves are characterized by the finite*

eigenvalues.

A trigonometric parameterization for scalar DDEs

Recall that the goal is to parameterize $\mathcal{T}$ with *simple exact formulas or expressions*. Naturally, for several special cases Theorem 3.14 can be simplified. We stress an important special case. The parameterization can be expressed in terms of trigonometric expressions if $n = 1$, i.e., if the DDE is scalar.

Corollary 3.17 *Let $A_k = a_k \in \mathbb{R}$, $k = 0, \ldots, m$ be the coefficients in the non-singular DDE* (3.1)*. Let*

$$\begin{aligned} h_k &= \frac{\varphi_k + 2p_k\pi}{\omega},\ k = 1, \ldots, m-1, \\ h_m &= \frac{\mp \text{sign } (a_m)\text{acos}\left(-\sum_{k=0}^{m-1} \frac{a_k}{a_m}\cos(\varphi_k),\right) + 2p_m\pi}{\omega}, \\ \omega &= -\sum_{k=0}^{m-1} a_k \sin(\varphi_k) \pm \sqrt{a_m^2 - \left(\sum_{k=0}^{m-1} a_k \cos(\varphi_k)\right)^2}, \end{aligned}$$

where $p_k \in \mathbb{Z}, k = 1 \ldots m$, $\varphi_k \in [-\pi, \pi], k = 1, \ldots, m-1$ and $\varphi_0 = 0$. Then $(h_1, \ldots, h_m) \in \mathcal{T}$. Furthermore, every point in $\mathcal{T}$ corresponds to (at least) one combination of $\varphi_1, \ldots, \varphi_{m-1}$, $p_1, \ldots, p_m$, i.e., the mapping,

$$h = (\varphi_1, \ldots, \varphi_{m-1}, p_1, \ldots p_m) \mapsto \{(h_1, \ldots, h_m)\}$$

is a parameterization

$$\mathcal{T} = \bigcup_{\varphi_k \in [-\pi,\pi], p_k \in \mathbb{Z}} h(\varphi_1, \ldots, \varphi_{m-1}, p_1, \ldots p_m).$$

Proof: For $n = 1$, (3.19) is a quadratic equation. We denote the quadratic equation (3.19), $a_m z^2 + 2bz + a_m = 0$. Clearly,

$$z = \frac{-b \pm \sqrt{b^2 - a_m^2}}{a_m},$$

which is of unit magnitude, i.e., $z \in \partial\mathbb{D}$ if and only if $|a| \geq |b|$. Since $b = \sum_{k=0}^{m-1} a_k \cos(\varphi_k)$, inserting into (3.20) yields

$$\omega = -\sum_{k=0}^{m-1} a_k \sin(\varphi_k) \pm \sqrt{a_m^2 - \left(\sum_{k=0}^{m-1} a_k \cos(\varphi_k)\right)^2}.$$

The proof is completed by noting that

$$-\text{Arg } z = -\text{atan}\begin{pmatrix}\pm\sqrt{b^2-a_m^2}/a_m\\ -b/a_m\end{pmatrix} = \mp\text{sign } (a_m)\text{acos}\left(-\frac{b}{a_m}\right)$$

where $\text{atan}\binom{a}{b}$ denotes the four-quadrant inverse tangent $\text{atan}\binom{a}{b} = \text{Arg } b + ia$. □

3.3.2 Plotting critical curves

We expect that the most common way to use the parameterization is to plot $\mathcal{T}$ with a computer. To make our results easily available it is hence worthwhile to describe the parameterization in an algorithmic way.

The pseudo-code to generate the critical curves for the two-delay DDE is given in Algorithm 3.1.

Algorithm 3.1 Plotting critical curves

INPUT: Stepsize Δ, matrices $A_0, \ldots, A_m$

OUTPUT: A list of pairs of critical delays (h_1, h_2)

1: for $\varphi = -\pi : \Delta : \pi$ do
2: Find eigenpairs (z_k, u_k) of (3.23)
3: for $k = 1 : \text{length}(z)$ do
4: if z_k is on unit circle then
5: Compute v_k such that $u_k = \text{vec } v_k v_k^*$
6: Compute $\omega_k = -iv_k^*\left(A_2 z_k + A_0 + A_1 e^{-i\varphi}\right) v_k$
7: Accept critical points (h_1, h_2)

$$\begin{aligned} h_1 &= \frac{\varphi + 2p\pi}{\omega_k},\ p = -p_{\max}, \ldots, p_{\max} \\ h_2 &= \frac{-\text{Arg } z_k + 2q\pi}{\omega_k},\ q = -p_{\max}, \ldots, p_{\max} \end{aligned}$$

8: end if
9: end for
10: end for

In Step 1, Δ is the stepsize of the parameter φ. The for-loop in Step 3 is over all the eigenvalues found in Step 2. In Step 5, the operation of determining

v_k from u_k can be done with a singular value decomposition as described after Lemma 3.15. However, even though the singular value decomposition is in a sense the most accurate approach, if the eigenvector u_k is determined to high accuracy it will be sufficient (for most cases) to pick any non-zero vector out the matrificiation of u_k, e.g. the first n entries of u_k. In Step 7, $p_{\max}$ is the number of branches which should be included in the computation. Step 7 is not computationally demanding. We can select $p_{\max}$ so large that the computation contains all relevant branches, say such that all delays smaller than some delay tolerance are found. This is possible because the delays are monotonically increasing or decreasing in the branch parameter. The generalization to more than two delays is straightforward. It involves a nesting of the outer iteration (step 1) with for-loops of the new free variables φ_k and computing the other delays in step 7 similar to h_1.

Note that, as we mentioned in Remark 3.3, the parameterization is not necessarily the most efficient way to plot $\mathcal{T}$ numerically. However, in order to have a useful result it should be possible to evaluate the parameterization efficiently. Hence, computational aspects of the parameterization are relevant, which is why we consider more efficient methods for the quadratic eigenvalue problem for moderate or large dimensions in Section 3.3.

3.3.3 Commensurate delays

In Section 3.3.1 we gave a parameterization for the set of critical delays $\mathcal{T}$ when the delays were independent. We now wish to present results similar to the parameterization in a different setting in order to provide further understanding to results in the literature, in particular the method by Chen, Gu, Nett [CGN95].

A DDE is called commensurate if all delays are integer multiples of some delay, say τ. That is, the delays are of the form $(h_1, \ldots, h_m) = (\tau n_1, \ldots, \tau n_m)$, $n_k \in \mathbb{Z}$, $\tau \in \mathbb{R}_+$. The ordered set of integers $(n_1, \ldots, n_m)$ defines a ray in delay-space, we now parameterize the critical delays along this ray, i.e., given $(n_1, \ldots, n_m)$ we characterize the subset of $\mathcal{T}$,

$$\{(h_1, \ldots, h_m) \in \mathcal{T} : h_k/h_j = n_k/n_j \in \mathbb{Q} \text{ or } n_j = 0 \,\text{for all}\, k, j \in \mathbb{Z}\}\,.$$

The set is parameterized in the following theorem.

Theorem 3.18 *Let Σ be a nonsingular DDE* (3.1) *with matrices A_k, $k = 0, \ldots, m$.*

Let $n_k \in \mathbb{Z}_+$ define a direction in delay-space, i.e., $(h_1, \ldots, h_m) = (\tau n_1, \ldots, \tau n_m)$, $\tau > 0$. The critical delays τ along this ray, i.e., $\mathcal{T} \bigcap (\cup_{\tau>0}(\tau n_1, \ldots, \tau n_m))$ are given by

$$(h_1, \ldots, h_m) = (\tau n_1, \tau n_2, \ldots, \tau n_m),$$

where

$$\tau = \frac{-\text{Arg } z + 2p\pi}{\omega},$$

for any $p \in \mathbb{Z}$ and for $v \in \mathbb{C}^n$, $v^*v = 1$, $\omega \in \mathbb{R}$, $z \in \partial D$ fulfilling

$$\sum_{k=0}^{m} \left(A_k v v^* z^{n_k} + v v^* A_k^T z^{-n_k} \right) = 0, \tag{3.25}$$

and

$$\omega = -iv^* \left(\sum_{k=0}^{m} A_k z^{n_k} \right) v, \tag{3.26}$$

where $n_0 = 0$ for notational convenience.

Proof: The proof is analogous to Theorem 3.14. □

Analogously to Section 3.3.1, if it is possible to find the solutions z and v of (3.25), then (3.26) yields ω and τ. Without loss of generality we let $n_m = \max_{k\in[1,\ldots,m]} n_k$. After vectorizing the matrix equation (3.25) we find that

$$\sum_{k=0}^{m} \left(I \otimes A_k z^{n_m+n_k} + A_k \otimes I z^{n_m-n_k} \right) u = 0 \tag{3.27}$$

where $u \in \mathbb{C}^{n^2}$ is the vectorization of vv^*, i.e., $u = \bar{v} \otimes v$. This equation is of the form

$$\sum_{k=0}^{N} B_k z^k u = 0, \tag{3.28}$$

which in the literature is known as a *polynomial eigenvalue problem*. Similar to quadratic eigenvalue problems, the most common way to solve polynomial eigenvalue problems is by companion linearization, which is analyzed and generalized in [MMMM06b] and [MMMM06a]. For instance, the eigenvalues of (3.28) are the eigenvalues corresponding to the generalized eigenvalue problem

$$z \begin{pmatrix} I & & & \\ & \ddots & & \\ & & I & \\ & & & B_N \end{pmatrix} w = \begin{pmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ -B_0 & \cdots & -B_{N-2} & -B_{N-1} \end{pmatrix} w, \tag{3.29}$$

where $w = (u^T, zu^T, z^2u^T, \cdots z^{N-1}u^T)^T$. By selecting B_k according to the coefficients in (3.27), we can compute all solutions z, v to (3.25) by solving the eigenvalue problem (3.29). Other methods, such as the *Jacobi-Davidson* method could be applied directly to the polynomial eigenvalue problem without linearizing it, as in e.g. [SBFvdV96].

From the approximate solution of (3.28), the Hermitian rank-one matrix approximating vv^* can be chosen similar to the choice in the previous section, i.e., the principal vector in the singular value decomposition, since the accuracy result in Lemma 3.15 generalizes to the polynomial case.

Lemma 3.19 *Let (z, x) be an eigenpair of the polynomial eigenvalue problem (3.28) and $x = \bar{v} \otimes v$, $\|x\| = 1$, $|z| = 1$. Say $\tilde{x} = x + y = u \otimes \bar{u} + q$ is an approximation of x, then the sine of the angle between the approximation u and the vector v is bounded by*

$$|\sin(u, v)| \leq \sqrt{\|y\| \left(1 + \sum_{k=0}^{N} \|B_k\|\right) + \|q\|}.$$

Proof: The proof is analogous to the proof of Lemma 3.15. □

Remark 3.20 *Theorem 3.18 combined with the companion linearization is very similar to the results in [CGN95]. If we let $h_k = hk$ and transpose the left terms in (3.27), the companion form (3.29) is the eigenvalue problem occurring in [CGN95]. The derivation in [CGN95] is based on a vectorized form directly. The transpose can be explained in our context by fact that we started with $L = M(s)X + XM(s)^*$ and $X = vv^*$. If we started with $M(s)X + X\overline{M(s)}$, $X = vv^T$, where w is the left eigenvector, i.e., $w^*M(s) = 0$ we get an additional transpose. We found this presentation slightly easier because of the symmetry of the eigenvector.*

3.3.4 Examples

We discuss some particularly interesting examples for the parameterization given in Theorem 3.14.

Example 3.21 (Classical) Even though the scalar single-delay DDE,

$$\dot{x}(t) = a_0 x(t) + a_1 x(t - h),$$

is well studied in the literature, it is illustrative to rederive classical results using Theorem 3.14. We will see that $\mathcal{T}$ is parameterized with only one integer p. The quadratic eigenvalue problem (3.22) is

$$z^2 a_1 + 2z a_0 + a_1 = 0$$

which has the solutions

$$z = \frac{-a_0 \pm \sqrt{a_0^2 - a_1^2}}{a_1} = \frac{-a_0 \pm i\sqrt{a_1^2 - a_0^2}}{a_1}.$$

The solution z is of unit magnitude if and only if $a_0^2 \leq a_1^2$. If this is not the case, there are no critical delays. The critical frequencies are given by (3.20)

$$\omega = -i(a_1 z + a_0) = \pm\sqrt{a_1^2 - a_0^2}.$$

From Theorem 3.14 we now have the critical delays

$$\begin{aligned} h &= -\frac{\operatorname{atan}\binom{\pm\operatorname{sgn}(a_1)\sqrt{a_1^2-a_0^2}}{-\operatorname{sgn}(a_1)a_0} + 2p\pi}{\pm\sqrt{a_1^2 - a_0^2}} \\ &= -\frac{\operatorname{sgn}(a_1)\operatorname{atan}\binom{\sqrt{a_1^2-a_0^2}}{-\operatorname{sgn}(a_1)a_0} \mp 2p\pi}{\sqrt{a_1^2 - a_0^2}}, \end{aligned}$$

where $\operatorname{atan}\binom{a}{b}$ denotes the four-quadrant inverse tangent $\operatorname{atan}\binom{a}{b} = \operatorname{Arg} b + ia$.

Using the identity

$$\operatorname{atan}\binom{\sqrt{a^2 - b^2}}{-\operatorname{sgn}(a)b} = \operatorname{acos}\left(-\frac{b}{a}\right)$$

we arrive at the final expression

$$h = \frac{-\operatorname{sgn}(a_1)}{\sqrt{a_1^2 - a_0^2}}\left(\operatorname{acos}\left(-\frac{a_0}{a_1}\right) + 2p\pi\right) \tag{3.30}$$

for any $p \in \mathbb{Z}$.

This equation is very similar to formulas found in the literature. See for instance [CG82] or [Nic01a, Section 3.4.1]. In the results in the literature, e.g. [Nic01a, Section 3.4.1] it is commonly assumed that $a_1 \leq -|a_0|$ as otherwise the differential equation resulting in the assumption that $h = 0$ is not stable. With this assumption a stability interval around $h = 0$ is given. We note that the more general formula for the critical delays (3.30) can be used to compute other stable intervals.

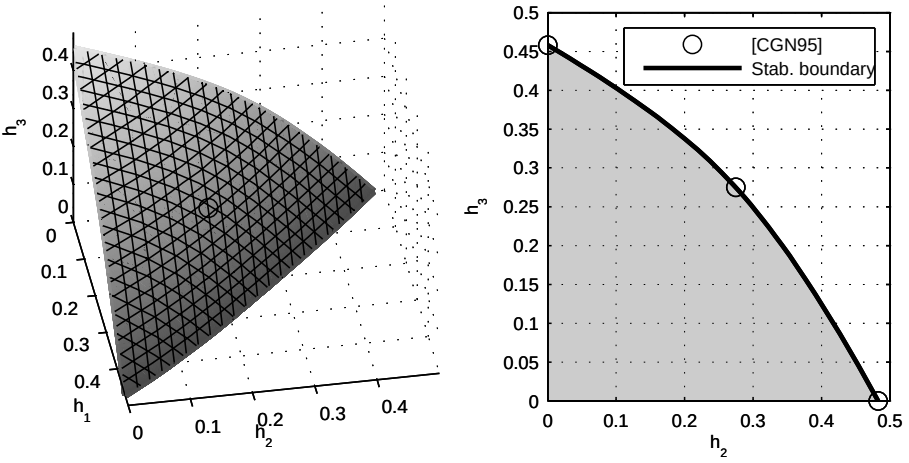
Example 3.22 The parameterization is applicable to DDEs with an arbitrary number of delays. It is however difficult to visualize $\mathcal{T}$ for $n > 3$. We here apply the parameterization (i.e., Algorithm 3.1) to the following DDE with three delays. Consider

$$\dot{x}(t) = A_0x(t) + A_1x(t-h_1) + A_2x(t-h_2) + A_3x(t-h_3)$$

where

$$A_0 = \begin{pmatrix} 3.8 & 0.3 \\ -0.1 & 4.6 \end{pmatrix}, \quad A_1 = \begin{pmatrix} -1.1 & 0 \\ 0 & -1.9 \end{pmatrix},$$
$$A_2 = \begin{pmatrix} -1.1 & 0.2 \\ 0.2 & -1.9 \end{pmatrix}, \quad A_3 = \begin{pmatrix} -1.9 & 0.4 \\ -0.2 & -1.0 \end{pmatrix}.$$

Since, the eigenvalues of $A_0 + A_1 + A_2 + A_3$ are all in the left half plane the origin in delay-space is stable and the first critical surface is the stability boundary. The first critical surface is plotted in Figure 3.4a. The critical curve for $h_1 = 0$ are plotted in Figure 3.4b. For verification, some special cases are computed with the method of Chen, Gu and Nett [CGN95] are given in both subfigures of Figure 3.4.



(a) (b) $h_1 = 0$

Figure 3.4: Critical surface and curves for Example 3.22

Example 3.23 We now apply the parameterization to a partial differential equation (PDE) with a delay. We first prove some theoretical properties of the PDE for some particular choices of the parameters without discretizing it. For the general case we discretize the PDE and apply the parameterization using Algorithm 3.1. We also wish to investigate how well the method scales to larger problems.

The following *partial delay-differential equation* (PDDE) belongs to a more general class of functions known as *partial functional differential equation* (PFDE). This type of equations in treated in the book [Wu96]. Consider

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + a_0(x)u + a_1(x)u(x,t-\tau_1) + a_2(x)u(x,t-\tau_2), \\ \qquad\qquad u(0,t) = u(\pi,t) = 0,\ t \geq 0 \end{cases} \tag{3.31}$$

where $a_0(x) = a_0 + \alpha_0 \sin(x)$, $a_1(x) = a_1 + \alpha_1 x(1 - e^{x-\pi})$ and $a_2(x) = a_2 + \alpha_2 x(\pi - x)$. We let $u = u(x,t)$ for notational convenience. This equation is a generalization of [Wu96, Example 1.12, Chapter 3]. We first discuss some exact analytic results and later treat a discretization of the PFDE. It is of interest to determine sufficient conditions on $a_0(x), a_1(x), a_2(x)$ such that the corresponding (delay-free) PDE, i.e., $\tau_1 = \tau_2 = 0$, is stable. Since the spectrum of the Laplace-operator with zero boundary conditions is $\{-k^2\}$, $k \in \mathbb{N}_+ \backslash \{0\}$, it follows from the min-max principle that all eigenvalues have negative real part if $a_0(x) + a_1(x) + a_2(x) < 1$ for all x. In particular, if $\alpha_1, \alpha_2 < 0, \alpha_0 > 0$ (which will be the case below) then it suffices that $a_0 + a_1 + a_2 + \alpha_0 - \alpha_1 \pi^2/4 < 1$.

For some cases, the critical delays can be computed explicitly. If $\alpha_0 = \alpha_1 = \alpha_2 = 0$ and $\tau_1 = \tau_2 = \tau$ we have that the spectrum is given by the solutions of

$$-\lambda + (a_0 - q^2) + (a_1 + a_2)e^{-\tau\lambda} = 0.$$

This follows from the separation of variables, i.e., if the coefficients in (3.31) are constant, then the equation can be separated in space and time by the ansatz, $u(x,t) = X(x)T(t)$, yielding the separated equations

$$X''(x) = (k - a_0)X(x),\ X(0) = X(\pi) = 0 \tag{3.32}$$

and

$$T'(t) = kT(t) + (a_1 + a_2)T(t - \tau). \tag{3.33}$$

Since the eigenvalues of the DDE (3.32) are $-q^2$, $q \in \mathbb{N}_+$ the feasible values of k are $k = a_0 - q^2$ and from the equation in T (3.33)

$$\lambda = (a_0 - q^2) + (a_1 + a_2)e^{-\tau\lambda}.$$

This corresponds to a scalar DDE and has a purely imaginary eigenvalue iff (cf. Example 3.21)

$$\tau = \frac{-\text{sign } (a_1 + a_2)}{\sqrt{(a_1 + a_2)^2 - (a_0 - q^2)^2}} \left(\text{acos} \left(\frac{q^2 - a_0}{a_1 + a_2} \right) + 2\pi p \right).$$

If $a_0(x)$ is not constant, then the same procedure can be applied with k as an eigenvalue of the operator corresponding to $\frac{\partial^2}{\partial x^2} + a_0$ with zero boundary conditions. However, if either a_1 or a_2 is not constant the problem can not be explicitly analyzed by separation of time and space in the same way. We wish to study this case which is not treatable with the separation of variables using a numerical scheme. We will hence treat the case where a_1 and a_2 have a small first derivative by discretization and verify that it is a small perturbation of the constant case.

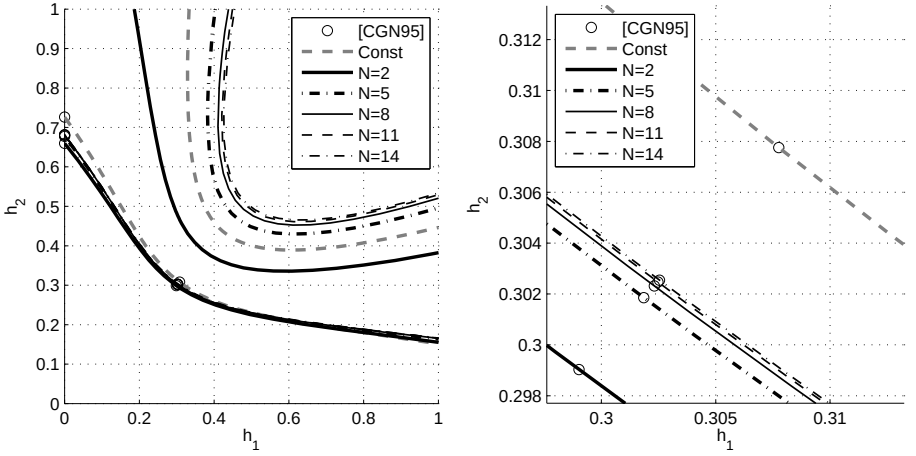
We discretize the PFDE with central difference and uniform stepsize $h = \pi/(N+1)$ in space and get the DDE of dimension N

$$\dot{v}(t) = \frac{(N+1)^2}{\pi^2} \begin{pmatrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{pmatrix} v(t) + \begin{pmatrix} a_0(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_0(x_N) \end{pmatrix} v(t) + \\ \begin{pmatrix} a_1(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_1(x_N) \end{pmatrix} v(t-h_1) + \begin{pmatrix} a_2(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_2(x_N) \end{pmatrix} v(t-h_2).$$

For the nonlinear case we pick $a_0 = 2$, $\alpha_0 = 0.3$, $a_1 = -2$, $\alpha_1 = 0.2$, $a_2 = -2$ and $\alpha_2 = -0.3$. The delay-free PFDE is stable since $a_0 + a_1 + a_2 + \alpha_0 - \alpha_2\pi^2/4 \approx -0.96 < 1$. The result of the parameterization of the critical curves for different number of nodes N is compared to the constant case $a_0 = 2.191$, $a_1 = -1.819$ and $a_2 = -2.509$ is shown in Figure 3.5. Some special cases can be computed with other methods. Note that the constant case can be solved analytically, since all matrices can be diagonalized. For instance, as mentioned, if $h_1 = h_2$, $a_0(x)$, $a_1(x)$ and $a_2(x)$ are constant then the (smallest) critical delay for the constant

case is

$$h_1 = h_2 = \frac{1}{\sqrt{(a_1+a_2)^2-(a_0-1)^2}}\left(\operatorname{acos}\left(\frac{1-a_0}{a_1+a_2}\right)\right) \approx 0.3105.$$



(a) Critical delays

(b) Critical delays, zoomed at the region where the delays are almost equal

Figure 3.5: Critical curves for Example 3.23 for different step-lengths

The points marked with circles in Figure 3.5 were computed with [CGN95]. The computational effort to compute one point using Algorithm 3.1 for the smallest stepsize $N = 14$, was in average 6.5 seconds. The free parameter was discretized with 63 points and the plot was produced in 6.9 minutes.

The figures indicate that for this particular PDDE, $N = 14$ discretization nodes is enough to determine the general geometry of the critical delays. This example will be treated for a finer discretization using a more efficient solver in Section 3.5.2.

3.4 A parameterization for neutral DDEs

The parameterization presented in the previous section can be generalized to *neutral DDEs*, i.e., DDEs with delayed derivative terms. Parts of these results were published in [Jar07]. We consider neutral DDEs of the type,

$$\Sigma = \begin{cases} \sum_{k=0}^{m} B_k \dot{x}(t-h_k) = \sum_{k=0}^{m} A_k x(t-h_k),\ t \geq 0 \\ x(t) = \varphi(t),\ t \in [-h_m, 0], \end{cases} \tag{3.34}$$

where we again let $h_0 = 0$ without loss of generality. The characteristic equation for (3.34) is

$$M(s)v := (-sB(s) + A(s))\, v = 0,\ ||v|| = 1, \tag{3.35}$$

where $A(s) = A_0 + \sum_{k=1}^{m} A_k e^{-h_k s}$ and $B(s) = B_0 + \sum_{k=1}^{m} B_k e^{-h_k s}$. As mentioned in the introduction, the spectrum is defined as $\sigma(\Sigma) := \{s \in \mathbb{C} : \det(M(s)) = 0\}$. In the literature, it is common to assume that $B_0 = I$. We will assume this in the description of some continuity and stability properties that follows. This is however not necessary for the parameterization of $\mathcal{T}$.

Discontinuity properties of neutral DDEs as well as sufficient conditions for continuity have been presented in [AH80] and more recently [MERD02]. We follow the notation and characterizations of continuity and stability in the book of Michiels and Niculescu [MN07b, Section 1.2].

In our context, neutral DDEs have two unpleasant properties which makes an analysis more involved. The origin can be a clustering point of the real part of the spectrum. Stability is hence only guaranteed if the supremum of the real part of the spectrum is negative, i.e., the spectral abscissa defined in (3.6) is negative, $\alpha(h_1, \ldots, h_m) < 0$. Unlike retarded DDEs, it is not sufficient for stability that the spectrum is contained in the open left half plane. Moreover, the spectral abscissa is not always continuous.

More precisely, the real part of the spectrum of neutral DDEs can have finite clustering points. Suppose $\{s_k\}$ is a sequence with $s_k \in \sigma(\Sigma)$ and $|\text{Im } s_k| \to \infty$. For the retarded case, this implies that Re s_k is unbounded, and Re $s_k \to -\infty$. However, for neutral DDEs, this limit can be finite, i.e., Re $s_k \to \zeta \in \mathbb{R}$. These limits ζ are the clustering points of the real part of the so-called difference operator corresponding to (3.34). Here the real part of the spectrum of the

associated difference operator is

$$Z_D = \left\{ \text{Re } s : \det \left(I + \sum_{k=1}^{m} B_k e^{-sh_k} \right) = 0 \right\}.$$

The clustering points of the real part of the spectrum of the neutral DDE is given by the closure of Z_D, i.e., $\zeta \in \operatorname{clos} Z_D$. For the retarded case $Z_D = \emptyset$, from which we verify that the real part of the spectrum of retarded DDEs have indeed no finite clustering points.

The finite clustering points of the real part of the spectrum are closely related to the fact that the spectral abscissa $\alpha(h_1, \ldots, h_m)$ is not always continuous. This important property unfortunately breaks the continuity argument as it implies that a stability switch does not necessarily mean that there is a purely imaginary eigenvalue. That is, $\partial D_s \subset \mathcal{T}$ does not hold in general.

Fortunately, there is theory available in the literature which gives sufficient conditions for a continuous spectral abscissa. In particular, if the associated difference operator is *strongly exponentially stable* then the spectral abscissa is continuous [MN07b, Theorem 1.39]. The associated difference operator is (defined as) strongly exponentially stable, if and only if

$$\max_{z \in \mathbb{Z}^k} r_\sigma \left(\sum_{k=1}^{m} B_k z_k \right) < 1, \tag{3.36}$$

where $r_\sigma(A)$ denotes the spectral radius of $A \in \mathbb{C}^{n \times n}$, i.e., $r_\sigma(A) = \max\{|s| : s \in \sigma(A)\}$. It is now important to note that (3.36) and hence the continuity of the spectral abscissa is *only* dependent on B_k, $k = 0, \ldots, m$ and not the delays, $h_1, \ldots, h_m$, nor the system matrices on right hand side, $A_0, \ldots, A_m$.

Note that the definition (3.36) is not stated such that it is easy to verify for a given example. In practice, the sufficient condition $\sum_{k=1}^{m} \|B_k\| < 1$ is often used.

Despite these somewhat pessimistic properties regarding the continuity of the spectrum and the spectral abscissa, our goal remains the same. We wish to construct a parameterization h expressed as computable expressions, mapping a simple mathematical object onto $\mathcal{T}$. Even though the discontinuity properties of the spectrum will prevent a stability analysis using a parameterization of $\mathcal{T}$ for the general case an explicit representation of $\mathcal{T}$ can still be useful in a practical analysis of the stability region in delay-space. Moreover, if the difference operator corresponding to the neutral part of (3.34) is strongly exponentially stable, e.g.

if $\sum_{k=1}^{m} \|B_k\| < 1$ then $\partial D_s \subset \mathcal{T}$ and the analysis can be carried out similar to the retarded case.

Lemma 3.24 *Suppose A, B and $M(s) = -sB(s) + A(s)$ are parameter dependent matrices and let*

$$\begin{aligned} L(X,s) &:= M(s)XB(s)^* + B(s)XM(s)^* = \\ &= A(s)XB(s)^* + B(s)XA(s)^* - 2B(s)XB(s)^* \mathrm{Re}\ s. \end{aligned} \tag{3.37}$$

For any $\omega \in \mathbb{R}$ and $v \in \mathbb{C}^n$, such that $v^*v = 1$ and $w := B(i\omega)v \neq 0$ the following are equivalent

$$M(i\omega)v = 0 \tag{3.38}$$

$$L(vv^*, i\omega) = 0 \ \wedge \ w^*M(i\omega)v = 0 \tag{3.39}$$

Proof: The implication (3.38) $\Rightarrow$ (3.39) is clear from the definition. For the implication (3.39) $\Rightarrow$ (3.38) we have

$$L(vv^*, s)w = M(s)vv^*B(s)^*w + B(s)vv^*M(s)^*w.$$

Clearly, since $v^*B(s)^*w = w^*w \neq 0$, (3.38) holds. □

Suppose $i\omega, v$ is an eigenpair of the DDE, then either $B(i\omega)v = 0$ or $B(i\omega)v \neq 0$. Many results for neutral DDEs are stated under mild conditions on the difference equation

$$B_0x(t) + B_1x(t-h_1) + \cdots B_mx(t-h_m) = 0. \tag{3.40}$$

As usual, the solutions $s \in \mathbb{C}$ of $\det(B(s)) = 0$ are called the eigenvalues of the difference equation. First note that, the condition $B(i\omega)v = 0$ means that the difference equation has a purely imaginary eigenvalue. Also note that if $B_0 = I$ then this will not occur if (3.40) is strongly exponentially stable. The difference equation (3.40) has a purely imaginary only for very special cases. We treat this separately in the main results that follow.

3.4.1 Main results

Similar to the retarded case, we identify some special cases which are easy to treat. This allows us to exclude the cases in the main theorem.

Lemma 3.25 *The following special cases hold for the neutral DDE* (3.34) Σ*:*

a) If $0 \in \sigma(\Sigma)$ *for some* $(h_1, \ldots, h_m)$ *then* $\mathcal{T} = \mathbb{R}_+^m$.

b) If the difference operator (3.40) *has a purely imaginary eigenvalue for some* $(h_1, \ldots, h_m)$*, i.e., there is* $v \in \mathbb{C}^n \backslash \{0\}$ *and* $\omega \in \mathbb{R}$ *such that* $0 = \sum_{k=0}^m B_k e^{-ih_k\omega} v$*,* $h_0 = 0$*, then* $(h_1, \ldots, h_m) \in \mathcal{T}$ *with corresponding eigenvector* v *and imaginary eigenvalue* $i\omega$ *if and only if* $0 = \sum_{k=0}^m A_k e^{-ih_k\omega} v$.

Proof: The proof of part a) is identical to the proof of Lemma 3.13.

It remains to show b). Suppose $v \in \mathbb{C}^n \backslash \{0\}$ and $\omega \in \mathbb{R}$ are such that $\left(\sum_{k=0}^m B_k e^{-ih_k\omega}\right) v = 0$. Then,

$$M(i\omega)v = \left(-i\omega \sum_{k=0}^m B_k e^{-ih_k\omega} + \sum_{k=0}^m A_k e^{-ih_k\omega} \right) v = \left(\sum_{k=0}^m A_k e^{-ih_k\omega} \right) v.$$

Hence, $(h_1, \ldots, h_m) \in \mathcal{T}$ with corresponding eigenvector v and eigenvalue $i\omega$ if and only if the right hand side is zero. □

Similar to the parameterization for retarded DDEs we now introduce the free variables $\varphi_k = h_k\omega$, $k = 1, \ldots, m-1$ and $z = e^{-ih_m\omega}$. To simplify the notation we denote $C(\vec{\varphi}) := \sum_{k=0}^{m-1} A_k e^{-i\varphi_k}$ and $D(\vec{\varphi}) := \sum_{k=0}^{m-1} B_k e^{-i\varphi_k}$, and introduce the parameterization vector $\vec{\varphi} = (\varphi_1, \ldots, \varphi_m)$ and $\varphi_0 = 0$.

Lemma 3.26 *Let* Σ *be a neutral DDE with* $0 \notin \sigma(\Sigma)$ *and fixed coefficients* $A_k, B_k \in \mathbb{R}^{n\times n}$*,* $k = 0, \ldots, m$*, such that the difference equation* (3.40) *does not have a purely imaginary eigenvalue. For any given* $\vec{\varphi} = (\varphi_1, \ldots, \varphi_{m-1}) \in [-\pi, \pi]^{m-1}$ *let* $z \in \partial\mathbb{D}$ *and* $v \in \mathbb{C}^n$ *be a solution to*

$$\begin{aligned} z^2 \quad & (A_m v v^* D(\vec{\varphi})^* + B_m v v^* C(\vec{\varphi})^*) + \\ z \quad & (C(\vec{\varphi}) v v^* D(\vec{\varphi})^* + A_m v v^* B_m^* + \\ & D(\vec{\varphi}) v v^* C(\vec{\varphi})^* + B_m v v^* A_m^*) + \\ & C(\vec{\varphi}) v v^* B_m^* + D(\vec{\varphi}) v v^* A_m^* = 0, \end{aligned} \tag{3.41}$$

and define

$$w := (B_m z + D(\vec{\varphi}))v. \tag{3.42}$$

Then $w \neq 0$ *and*

$$\omega := -i \frac{w^* \left(A_m z + C(\vec{\varphi})\right) v}{w^* w} \in \mathbb{R}. \tag{3.43}$$

Proof: The assumption is that the difference equation does not have a purely imaginary eigenvalue, i.e., $B_0v+\sum_{k=1}^{m} B_kz_kv \neq 0$ for all $z_1,\ldots,z_m \in \partial\mathbb{D}$. Hence, $w \neq 0$.

Note that since $z \in \partial\mathbb{D}$, (3.41) can be written as

$$Pvv^*Q^* + Qvv^*P^* = 0, \tag{3.44}$$

where $P = A_mz + C(\vec{\varphi})$ and $Q = B_mz + D(\vec{\varphi})$. With this notation $w = Qv$. Since, $w^*w \in \mathbb{R}$, in order to have $\omega \in \mathbb{R}$ it remains to show that $w^*Pv \in i\mathbb{R}$. If we multiply (3.44) with w^* from the left and w from the right, then

$$w^*Pvw^*w + w^*wv^*P^*w = 0.$$

Since $w^*w \neq 0$ by (3.42) it follows that $0 = w^*Pv + v^*P^*w = w^*Pv + (w^*Pv)^*$ and $w^*Pv \in i\mathbb{R}$. □

Lemma 3.27 *Let Σ be as in Lemma 3.26. For any $(\varphi_1,\ldots,\varphi_{m-1}) \in [-\pi,\pi]^{m-1}$ let $z \in \partial\mathbb{D}$ and $v \in \mathbb{C}^n$ be a solution to* (3.41)*, $w \neq 0$ as in* (3.42) *and $\omega \in \mathbb{R}$ be defined by* (3.43)*. Moreover, let*

$$h_k = \frac{\varphi_k + 2p_k\pi}{\omega},\ k = 1,\ldots,m-1, \tag{3.45}$$

$$h_m = \frac{-\text{Arg } z + 2p_m\pi}{\omega}. \tag{3.46}$$

For any $p_0,\ldots,p_m$, the point in delay-space $(h_1,\ldots,h_m)$ is a critical delay, i.e., $(h_1,\ldots,h_m) \in \mathcal{T}$.

Proof: We will show that the construction of $v \in \mathbb{C}^n$, ω, z, $h_1,\ldots,h_m$, i.e., (3.41), (3.43), (3.45) and (3.46), yields that $M(i\omega)v = 0$. From Lemma 3.24 it is sufficient to show that $L(vv^*,i\omega) = 0$,$w \neq 0$ and $w^*M(i\omega)v = 0$. First note that $e^{-i\varphi_k} = e^{-ih_k\omega}$, $k = 1,\ldots,m-1$ from (3.45) and $z = e^{-ih_m\omega}$ from (3.46). Hence, (3.41) is exactly $zL(vv^*,i\omega) = 0$. Since $z \neq 0$, $L(vv^*,i\omega) = 0$. Moreover, by rearranging the terms of (3.43), we have $w^*A(i\omega)v - i\omega w^*w = 0$. Hence, the conditions (3.39) are fullfilled and $M(i\omega)v = 0$ for this construction of $v,\omega,h_1,\ldots,h_m$. □

Lemma 3.28 *Let Σ be as in Lemma 3.26. If $(h_1,\ldots,h_m) \in \mathcal{T}$ then there exist $(\varphi_1,\ldots,\varphi_{m-1}) \in [-\pi,\pi]^{m-1}$ and $p_0,\ldots,p_m$ such that* (3.41)*,* (3.42)*,* (3.43)*,* (3.45) *and* (3.46) *are fulfilled.*

Proof: We show that for any $(h_1, \ldots, h_m) \in \mathcal{T}$ (with eigenvalue $i\omega$ and eigenvector v) there is a $\varphi_1, \ldots, \varphi_{m-1}$ such that the construction of v, ω, z yields exactly this $(h_1, \ldots, h_m)$.

From Lemma 3.24 it is clear that $L(vv^*, i\omega) = 0$ and $w^*M(i\omega)v = 0$. Let $z = e^{-ih_m\omega}$ and p_m such that (3.46) is fulfilled. Similarly we pick $\varphi_k = \omega h_k + 2p_k\pi$ such that $\varphi_k \in [-\pi, \pi]$ and $p_k \in \mathbb{Z}$ and (3.46) holds. Note that since $L(vv^*, i\omega) = 0$, $zL(vv^*, i\omega) = 0$ and hence (3.41) also holds. Finally, (3.43) follows from (3.39), i.e., the fact that $w^*M(i\omega)v = 0$. The constructed $\varphi_1, \ldots, \varphi_{m-1}$ yield the given $h_1, \ldots, h_m$. □

The two lemmas imply that the construction is a parameterization.

Theorem 3.29 *Let Σ be as in Lemma 3.26. The map*

$$(\varphi_1, \ldots, \varphi_{m-1}) \mapsto \bigcup_{p \in \mathbb{Z}^m} (h_1, \ldots, h_m)$$

where $h_1, \ldots, h_m$ are defined by (3.45) *and* (3.46) *is a parameterization of $\mathcal{T}$.*

Proof: This follows directly from the forward implication Lemma 3.27 and the backward implication Lemma 3.28. □

Similar to the previous section we note that (3.41) is a matrix equation which can be vectorized into

$$(z^2M(\vec{\varphi}) + zG(\vec{\varphi}) + K(\vec{\varphi}))u = 0, \tag{3.47}$$

where $u = \operatorname{vec} vv^*$, $M(\vec{\varphi}) = D(-\vec{\varphi}) \otimes A_m + C(-\vec{\varphi}) \otimes B_m$, $G(\vec{\varphi}) = D(-\vec{\varphi}) \otimes C(\vec{\varphi}) + B_m \otimes A_m + C(-\vec{\varphi}) \otimes D(\vec{\varphi}) + A_m \otimes B_m$ and $K(\vec{\varphi}) = B_m \otimes C(\vec{\varphi}) + A_m \otimes D(\vec{\varphi})$. Equation (3.47) is a polynomial eigenproblem of degree two, i.e., a quadratic eigenproblem. Again, the solutions of the quadratic eigenproblems can be computed from a corresponding companion form, e.g.

$$z \begin{pmatrix} I & 0 \\ 0 & M(\vec{\varphi}) \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix} = \begin{pmatrix} 0 & I \\ -K(\vec{\varphi}) & -G(\vec{\varphi}) \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix}. \tag{3.48}$$

We can hence compute the critical delays for neutral DDEs analogously to Algorithm 3.1.

At the cost of technicality, the scalar parameterization in Corollary 3.17 can be generalized to neutral DDEs.

Corollary 3.30 *Let $A_k = a_k \in \mathbb{R}$, $B_k = b_k$, $k = 0, \ldots, m$ be the coefficients in the nonsingular DDE* (3.4)*. The mapping $h = (\varphi_1, \ldots, \varphi_{m-1}, p_1, \ldots, p_m) \mapsto \{(h_1, \ldots, h_m)\}$ is a surjective mapping onto $\mathcal{T}$ where*

$$\begin{aligned}
h_k &= \frac{\varphi_k + 2p_k\pi}{\omega},\ k = 1, \ldots, m-1 \\
h_m &= \frac{1}{\omega(\vec{\varphi})}\left(\text{Arg}\ \left(a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})\right) \mp \text{acos}\left(-\frac{a_m b_m + \text{Re}\ \left(b(\vec{\varphi})a(\vec{\varphi})^*\right)}{|a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})|}\right)\right. \\
&\qquad \left. + 2p_m\pi\right) \\
\omega(\vec{\varphi}) &= \frac{a_m \text{Im}\ z(\vec{\varphi}) + \text{Im}\ a(\vec{\varphi})}{b_m \text{Re}\ z(\vec{\varphi}) + \text{Re}\ b(\vec{\varphi})} \\
z(\vec{\varphi}) &= \frac{-\alpha \pm \sqrt{(a_m b_m + \text{Re}\ \left(b(\vec{\varphi})a(\vec{\varphi})^*\right))^2 - |a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})|^2}}{a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})}.
\end{aligned}$$

where $p_k \in \mathbb{Z}, k = 1 \ldots m$, $\varphi_k \in [-\pi, \pi], k = 1, \ldots, m-1$, $a(\vec{\varphi}) = \sum_{k=0}^m a_k e^{-i\varphi_k}$, $b(\vec{\varphi}) = \sum_{k=0}^m b_k e^{-i\varphi_k}$ and $\varphi_0 = 0$.

Proof: For $n = 1$, (3.41) is a quadratic equation. We temporarily introduce $q = a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi}) \in \mathbb{C}$ and $2\alpha = a(\vec{\varphi})b(\vec{\varphi})^* + 2a_m b_m + b(\vec{\varphi})a(\vec{\varphi})^* \in \mathbb{R}$. The quadratic equation (3.41) is now

$$z^2 q + 2z\alpha + q^* = 0,$$

i.e.,

$$z = \frac{-\alpha \pm \sqrt{\alpha^2 - qq^*}}{q}. \tag{3.49}$$

There are critical delays if and only if $z \in \partial\mathbb{D}$, i.e., $|q| \geq |\alpha|$. Now let $q = re^{i\theta}$, $r > 0$. Hence,

$$\begin{aligned}
-\text{Arg}\ z = \theta \mp \text{atan}\begin{pmatrix}\sqrt{r^2 - \alpha^2} \\ -\alpha\end{pmatrix} = \theta \mp \text{acos}\left(-\frac{\alpha}{r}\right) = \\
= \text{Arg}\ \left(a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})\right) \mp \text{acos}\left(-\frac{a_m b_m + \text{Re}\ \left(b(\vec{\varphi})a(\vec{\varphi})^*\right)}{|a_m b(\vec{\varphi})^* + b_m a(\vec{\varphi})|}\right)
\end{aligned} \tag{3.50}$$

We now find an explicit expression for ω. Clearly,

$$\omega(b_m z + b) = -i(a_m z + a)$$

and

$$\omega = \frac{a_m \text{Im } z(\vec{\varphi}) + \text{Im } a(\vec{\varphi})}{b_m \text{Re } z(\vec{\varphi}) + \text{Re } b(\vec{\varphi})}.$$

□

3.4.2 Commensurate delays

Following the same ideas as in Section 3.3.3, we now assume that the delays are integer multiples of some delay h, i.e., $h_j = hj$ for $j = 0, \ldots, m$. We substitute $\zeta = e^{-i\omega h}$, and rephrase the theorem. We also note that these results are similar to the results of the method by Fu, Niculescu and Chen [FNC06].

Theorem 3.31 *Consider the nonsingular, commensurate m delay DDE with the corresponding difference equation* (3.40) *which does not have a purely imaginary eigenvalue. The DDE has the critical delay h, if and only if there is an $\zeta \in \partial\mathbb{D}$, $\omega \in \mathbb{R}$ and $v \in \mathbb{C}^n$ such that*

$$\sum_{j,k}^{m} \left(A_j v v^* B_k^T + B_j v v^* A_k^T \right) \zeta^{m+j-k} = 0 \tag{3.51}$$

and

$$\omega = -i \frac{w^* \sum_j^m A_j \zeta^j v}{w^* w}$$

and

$$h = \frac{-\text{Arg } \zeta + 2p\pi}{\omega}$$

for some $p \in \mathbb{Z}$, where

$$w = \sum_{j}^{m} B_j \zeta^j v.$$

Proof: This follows from Lemma 3.24, if we let $\zeta = e^{-i\omega h}$. □

We now note that (3.51) can be vectorized, i.e., stacking the columns of the matrix-equation on top of each other, into

$$\sum_{j,k=0}^{m} \left(B_k \otimes A_j + A_k \otimes B_j\right) \zeta^{m+j-k} u = \sum_{q=0}^{2m} \sum_{j=\max(q-m,0)}^{\min(m,q)} \left(B_{j-q+m} \otimes A_j + A_{j-q+m} \otimes B_j\right) \zeta^q u \quad (3.52)$$

where $u = \operatorname{vec} vv^*$. This is a polynomial eigenvalue problem for which numerical methods exist.

For instance the companion form of the polynomial eigenvalue problem

$$\sum_{k=0}^{N} C_k \zeta^k u = 0,$$

fulfills the equation

$$\zeta \begin{pmatrix} I & & & \\ & \ddots & & \\ & & I & \\ & & & C_N \end{pmatrix} w = \begin{pmatrix} 0 & I & & \\ & \ddots & \ddots & \\ & & 0 & I \\ -C_0 & \cdots & -C_{N-2} & -C_{N-1} \end{pmatrix} w, \quad (3.53)$$

where $w = (u^T, \zeta u^T, \zeta^2 u^T, \cdots \zeta^{N-1} u^T)^T$, which is a generalized eigenvalue problem, solvable on a computer (for moderate sized problem) to sufficient accuracy.

For the polynomial eigenvalue problem (3.52), we have $N = 2m$ and

$$C_q = \sum_{j=\max(q-m,0)}^{\min(m,q)} B_{j-q+m} \otimes A_j + A_{j-q+m} \otimes B_j$$

for $q = 0, \ldots, 2m$.

Remark 3.32 *A similar matrix condition is contained in [FNC06]. Note that the matrices Q_k in [FNC06] are not identical to C_q. Similar to the relation in Remark 3.20 the eigenvalue problems are identical apart from transposes.*

3.4.3 Examples

To demonstrate the value of Theorem 3.29 we consider some interesting special cases. We derive some explicit expression which are to our knowledge not known,

and show by example how the parameterization can be used to numerically compute the critical curves (Example 3.37).

Example 3.33 ([FNC06]) For the neutral time-delay system

$$\dot{x}(t) + b_1\dot{x}(t-h) = a_0x(t) + a_1x(t-h)$$

we have that $B(i\omega) = 1 + b_1e^{-ih\omega}$ and $A(i\omega) = a_0 + a_1e^{-ih\omega}$. If we assume that $b_1 \neq 0$ and $a_1 \neq b_1a_0$ then $A(i\omega) = 0$ and $B(i\omega) = 0$ and we can apply Theorem 3.29. Moreover, we have that $C(\vec{\varphi}) = a_0$,$D(\vec{\varphi}) = 1$, $A_m = a_1$ and $B_m = b_1$. Even though the mapping can be found by simply inserting A and B into Corollary 3.30 it is illustrative to perform the derivation using Theorem 3.29. The quadratic eigenvalue problem (3.47) is now

$$z^2(a_1 + b_1a_0) + 2z(a_0 + a_1b_1) + a_0b_1 + a_1 = 0$$

which has the solutions

$$z = \frac{-(a_0 + a_1b_1) \pm \sqrt{(a_0 + a_1b_1)^2 - (a_0b_1 + a_1)^2}}{a_0b_1 + a_1} = \\ = \frac{-(a_0 + a_1b_1) \pm i\sqrt{(a_0b_1 + a_1)^2 - (a_0 + a_1b_1)^2}}{a_0b_1 + a_1}. \quad (3.54)$$

The DDE has critical delays if and only if $(a_0b_1 + a_1)^2 > (a_0 + a_1b_1)^2$, which implies that $b_1 \neq 1$. Note that if $|z| = 1$ then

$$-\text{Arg } (z) = \mp\text{sign } (a_0b_1 + a_1)\text{acos}\left(-\frac{a_0 + a_1b_1}{a_0b_1 + a_1}\right).$$

The imaginary eigenvalues are $i\omega$, where

$$\omega = \frac{a_0 + a_1z}{i(1 + b_1z)} = -\frac{a_0 + a_1\text{Re } z}{b_1\text{Im } z} = \pm\frac{a_1^2 - a_0^2}{\sqrt{(a_0b_1 + a_1)^2 - (a_0 + a_1b_1)^2}} = \\ \pm \text{ sign } (a_1^2 - a_0^2)\frac{\sqrt{(a_1^2 - a_0^2)^2}}{\sqrt{(b_1^2 - 1)(a_0^2 - a_1^2)}} = \mp\text{sign } (a_0^2 - a_1^2)\sqrt{\frac{a_1^2 - a_0^2}{1 - b_1^2}}. \quad (3.55)$$

Finally, the critical delays are now given by Theorem 3.29

$$h = \frac{-\text{Arg } z + 2p\pi}{\omega} = \rho\sqrt{\frac{1 - b_1^2}{a_1^2 - a_0^2}}\left(\text{acos}\left(-\frac{a_0 + a_1b_1}{a_0b_1 + a_1}\right) + 2p\pi\right), \quad (3.56)$$

where $\rho = \operatorname{sgn}\left(\frac{a_0^2 - a_1^2}{a_0 b_1 + a_1}\right)$. Note that the signs of $\pm$ were matched and canceled. Also note that if $b_1 = 0$, i.e., the retarded case (cf. (3.30)), $\rho = -\operatorname{sgn}(a_1)$ as expected, because $a_1^2 \geq a_0^2$ if there are any critical delays.

Even though single delay scalar neutral DDEs have been extensively studied, we have not seen this expression for the critical delays in the literature.

There are however very similar expressions in the literature. We point out the minor differences. In the book of Niculescu [Nic01a, Proposition 3.16] the first stability interval including $h = 0$ is expressed exactly using atan. We find the parameterization (3.56) slightly simpler than the expression of Niculescu. Moreover, (3.56) can be used to determine stability intervals not necessarily including $h = 0$. The expression for ω (sometimes called switching frequency or crossing frequency) is contained in [FNC06]. A formula for the critical delays expressed as the angle of a complex number is given in the same article. This is expected to be a different form of (3.56).

Example 3.34 For the neutral two-delay system

$$b_1 \dot{x}(t - h_1) + b_2 \dot{x}(t - h_2) = x(t)$$

we have that $C(\varphi) = 1$, $A_m = 0$, $B(\varphi) = e^{-i\varphi}$, $B_m = b_2$. The case that the difference operator has a purely imaginary eigenvalue can be excluded by the fact that $A(s) = 1 \neq 0$. The quadratic eigenproblem corresponding to (3.47) is

$$z^2 b_2 + z 2 b_1 \cos(\varphi) + b_2 = 0$$

and

$$z = \frac{-b_1 \cos(\varphi) \pm \sqrt{b_1^2 \cos^2(\varphi) - b_2^2}}{b_2} = \frac{-b_1 \cos(\varphi) \pm i\sqrt{b_2^2 - b_1^2 \cos^2(\varphi)}}{b_2}. \quad (3.57)$$

For the parametrization to yield proper critical delays we require that φ fulfills $b_2^2 \geq b_1^2 \cos^2(\varphi)$. The frequencies ω are

$$\omega = \frac{1}{i(b_1 e^{-i\varphi} + (-b_1 \pm i\sqrt{b_2^2 - b_1^2 \cos^2(\varphi)}))} = \frac{1}{b_1 \sin(\varphi) \mp \sqrt{b_2^2 - b_1^2 \cos^2(\varphi)}}. \quad (3.58)$$

Hence, a parametrization of the critical delays is given by

$$\begin{aligned} h_1 &= \frac{\varphi + 2p\pi}{\omega} = \left(b_1 \sin(\varphi) \mp \sqrt{b_2^2 - b_1^2 \cos^2(\varphi)}\right)(\varphi + 2p\pi) \\ h_2 &= \frac{-\text{Arg } z + 2q\pi}{\omega} = \\ &\left(b_1 \sin(\varphi) \mp \sqrt{b_2^2 - b_1^2 \cos^2(\varphi)}\right)\left(2q\pi \pm \text{sgn}(b_2)\text{acos}\left(-\frac{b_1 \cos(\varphi)}{b_2}\right)\right), \end{aligned} \tag{3.59}$$

for any $p, q \in \mathbb{Z}$.

Example 3.35 We now consider the neutral system corresponding to Example 3.33 where the two delays are not necessarily equal, i.e.,

$$\dot{x}(t) + b_1 \dot{x}(t - h_1) = a_0 x(t) + a_2 x(t - h_2).$$

We have that $b(\varphi) = 1 + b_1 e^{-i\varphi}$, $b_m = 0$, $a(\varphi) = a_0$ and $a_m = a_2$. The quadratic eigenproblem/equation corresponding to (3.47) is

$$z^2 a_2 (1 + b_1 e^{i\varphi}) + 2z a_0 (1 + b_1 \cos(\varphi)) + a_2 (1 + b_1 e^{-i\varphi}) = 0.$$

After many simple manipulations, which we leave out for brevity, we arrive at an expression for the critical frequencies.

$$\omega(\varphi) = \frac{a_0 + a_2 \text{Re } z}{b_1 \sin(\varphi)} = \tag{3.60}$$

$$\begin{aligned} &= \frac{a_0}{b_1 \sin(\varphi)} - \frac{a_0}{b_2 \sin(\varphi)} + \\ &\frac{a_0 b_1 \sin(\varphi)}{1 + 2b_1 \cos(\varphi) + b_1^2} + \\ &\mp \frac{\sqrt{a_2^2(1 + b_1^2 + 2b_1 \cos(\varphi)) - a_0^2(1 + b_1 \cos(\varphi))^2}}{1 + 2b_1 \cos(\varphi) + b_1^2} \end{aligned} \tag{3.61}$$

It is clear that even for examples like this, which may seem simple, the explicit real trigonometric expression (3.61) is too large to easily identify properties of the critical frequencies. Moreover, the complex expression (3.60) might be to prefer from a computational point of view as it requires less operations.

For brevity we only express the critical delays using the complex expression,

$$\begin{aligned} h_1 &= \frac{\varphi + 2p\pi}{\omega(\varphi)} = \frac{(b_1 \sin(\varphi))(\varphi + 2p\pi)}{a_0 + a_2 \text{Re } z}, \\ h_2 &= \frac{-\text{Arg } z + 2q\pi}{\omega(\varphi)} = \frac{(b_1 \sin(\varphi))(-\text{Arg } z + 2q\pi)}{a_0 + a_2 \text{Re } z}. \end{aligned}$$

Example 3.36 (From [IND+03] and [RKDD04]) With this example we show how one can find the critical delays of some multi-dimensional systems analytically. The commonly occurring example,

$$\dot{x}(t) - 0.1\dot{x}(t-h_1) = \begin{pmatrix} -2 & 0 \\ 0 & -0.9 \end{pmatrix} x(t) + \begin{pmatrix} -1 & 0 \\ -1 & -1 \end{pmatrix} x(t-h_2),$$

can be decoupled because all matrices are triangular. The spectrum is hence the union of the spectrum of the two decoupled systems

$$\begin{aligned} \dot{y}_1(t) - 0.1\dot{y}_1(t-h_1) &= -2y_1(t) - y_1(t-h_1), \\ \dot{y}_2(t) - 0.1\dot{y}_2(t-h_2) &= -0.9y_2(t) - y_2(t-h_2). \end{aligned}$$

If we let $h_1 = h_2 =: h$ we can apply the result of Example 3.33 (or [FNC06]). Here the system corresponding to y_1 does not have any critical delays. For y_2 we have $a_0 = -0.9$, $b_0 = 1$, $b_1 = -0.1$, $a_1 = -1$. Hence, $\rho = 1$. From (3.56) the critical delays are

$$h = \sqrt{\frac{1-0.1^2}{1-0.9^2}} \left(\operatorname{acos}\left(\frac{-0.8}{0.91}\right) + 2p\pi \right) = 3\sqrt{\frac{11}{19}} \left(\operatorname{acos}\left(\frac{-80}{91}\right) + 2p\pi \right), \quad (3.62)$$

which is an exact expression. For $p = 0$ we have $h \approx 6.0372$.

For the case that $h_1 \neq h_2$ we can apply the formula from Example 3.35 (or directly apply the numerical scheme Algorithm 3.1) to produce the critical curves. The resulting critical curves are shown in Figure 3.6. The delay margin is the smallest delay which a system turns unstable. The delay margin for this example, e.g. [IND+03] is also plotted.

Example 3.37 As a last example we apply the numerical scheme to an example for which we believe there is no simple analytical expression.

We now consider a modification of the example in [MV05, Section 5]. The example involves stabilization of a time-delay system using delayed feedback. We set the delay in the feedback to zero and investigate stability with respect to the closed stabilized system. The DDE is

$$\dot{x}(t) + B_1\dot{x}(t-h_1) + B_2\dot{x}(t-h_2) = A_0x(t),$$

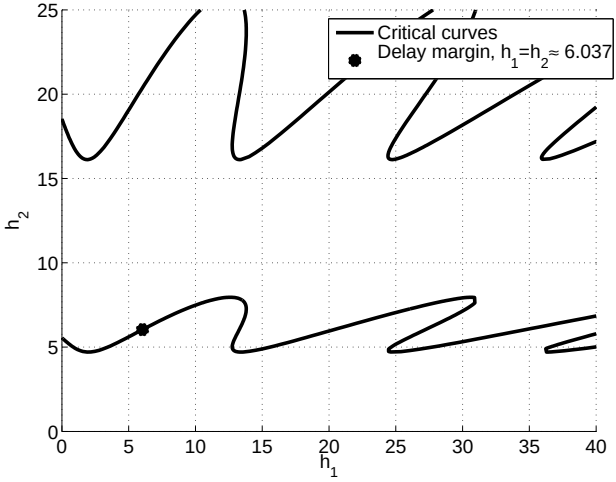


Figure 3.6: Critical curves for Example 3.36

where

$$
\begin{aligned}
B_1 &= -\begin{pmatrix} 0 & 0.2 & -0.4 \\ -0.5 & 0.3 & 0 \\ 0.2 & 0.7 & 0 \end{pmatrix}, B_2 = -\begin{pmatrix} -0.3 & -0.1 & 0 \\ 0 & 0.2 & 0 \\ 0.1 & 0 & 0.4 \end{pmatrix} \\
A_0 &= \begin{pmatrix} -4.8 & 4.7 & 3 \\ 0.1 & 1.4 & -0.4 \\ 0.7 & 3.1 & -1.5 \end{pmatrix} + BK^T, \\
B &= \begin{pmatrix} 0.3 & 0.7 & 0.1 \end{pmatrix}^T, K = \begin{pmatrix} -2.593 & 1.284 & 1.826 \end{pmatrix}^T.
\end{aligned}
$$

Note that A_0 is not singular, and hence $A(s)v = A_0 v = 0$ has no solutions and we can apply Theorem 3.29. The critical curves are plotted in Figure 3.7.

3.5 Solving the quadratic eigenproblem

The eigenvalue problems (3.22) and (3.47) are vectorizations of matrix equations of size $2n^2 \times 2n^2$ and can hence be of large size, even if the dimension n of the

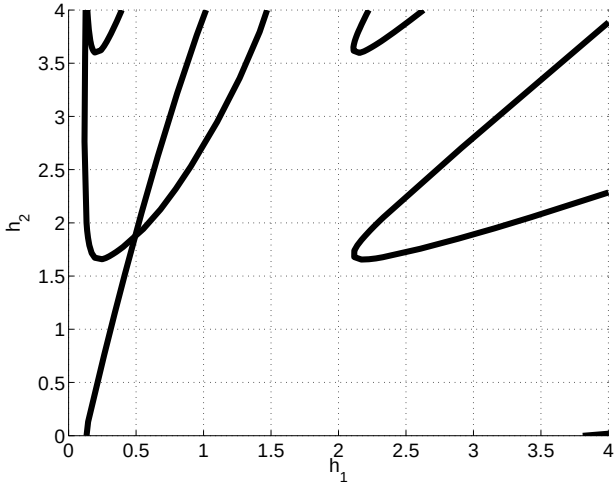


Figure 3.7: Critical curves for Example 3.37

DDE is moderate[3]. More precisely, to compute all eigenvalues of a full $N \times N$ matrix, a computational effort essentially proportional to N^3 is required. Here $N = 2n^2$ and the computational effort is proportional to n^6. Since this is likely to be the computationally dominating part of the parameterization, we now wish to show that it is possible to improve the efficiency by exploiting the structure of the matrices. The efficiency is improved by using the fact that the shift-invert operation of the companion matrix can be computed from the solution of a Lyapunov-equation.

An analogous approach was taken by the author in [Jar06b], but with the Cayley-transformed system.

Note that we here use the companion linearization even though there are linearizations (3.22) which preserves the structure. We mentioned in Section 3.3.1 that structured linearizations are likely to have better numerical stability properties. Moreover, a structured linearization may allow the construction of a

[3]On a current desktop computer (say AMD 2.2. GHz) moderate here means less than 100. Eigenvalue problems of order 2000 can be solved in a matter of minutes, hence, $n = 31 \approx \sqrt{1000}$ can be treated. However, eigenvalues of full matrices with dimension $N = 20000$, i.e., $n = 100$, can not be computed on a desktop computer with standard software in "adequate" time.

more efficient numerical method, if, for instance, a structure-preserving similarity transformations can be constructed. In particular, a structured linearization of the quadratic eigenvalue problem considered here is discussed in [FMMS07].

The idea to exploit the structure of matrices to compute fast inverse and shift-and-inverse operations was also used in a method to efficiently compute the distance to uncontrollability (of a dynamical system) in [GMO+06]. In that application, the computational complexity was reduced from $\mathcal{O}(n^6)$ to $\mathcal{O}(n^4)$.

3.5.1 Exploiting the structure

The first companion form of the quadratic eigenvalue problem (3.22) is

$$\begin{pmatrix} 0 & I \\ K & G \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix} = z \begin{pmatrix} I & 0 \\ 0 & -M \end{pmatrix} \begin{pmatrix} u \\ zu \end{pmatrix},$$

where

$$M = I \otimes A_m, G = \left(\sum_{k=0}^{m-1} I \otimes A_k e^{-i\varphi_k} + A_k \otimes I e^{i\varphi_k} \right), K = A_m \otimes I.$$

For simplicity we will assume that M is non-singular and that $m = 1$. In an iterative method for eigenvalue problems an efficient matrix-vector product can often be exploited to improve efficiency of one iteration. The traditional approach is to compute the LU-decomposition of the matrix before starting the iteration and solve two triangular systems in each matrix vector product. For instance, in standard procedures such as shift-and-inverted Arnoldi, the matrix vector product $(A - \sigma I)^{-1}x$ must be solved for many right hand sides x. For the shift-and-inverted Arnoldi, the LU-decomposition of $A - \sigma I$ is computed before the iteration starts, i.e., $A - \sigma I = LU$ and the two triangular systems $U^{-1}(L^{-1}x)$ are solved in each iteration. Linear systems where the matrix is triangular can be solved efficiently with forward or bachward substitution.

It turns out that we can produce a method which is even faster that the standard LU-decomposition-approach by exploiting the structure of this problem.

We wish to construct an efficient shift-invert operation of the matrix

$$A = \begin{pmatrix} I & 0 \\ 0 & -M \end{pmatrix}^{-1} \begin{pmatrix} 0 & I \\ K & G \end{pmatrix},$$

i.e., the operation

$$x = (A - \sigma I)^{-1} b. \tag{3.63}$$

The shift invert operation (3.63) is equivalent to

$$b = (A - \sigma I)x$$

and

$$\begin{pmatrix} I & 0 \\ 0 & -M \end{pmatrix} b = \left(\begin{pmatrix} 0 & I \\ K & G \end{pmatrix} - \sigma \begin{pmatrix} I & 0 \\ 0 & -M \end{pmatrix} \right) x.$$

Hence,

$$x = -\begin{pmatrix} (\sigma^2 M + \sigma G + K)^{-1} & 0 \\ 0 & (\sigma^2 M + \sigma G + K)^{-1} \end{pmatrix} \begin{pmatrix} G + \sigma M & M \\ -K & \sigma M \end{pmatrix} b = \frac{1}{\sigma} \begin{pmatrix} L^{-1}((G + \sigma M)b_1 + M b_2) \\ L^{-1}(-K b_1 + \sigma M b_2) \end{pmatrix},$$

where $L = \sigma M + G + \sigma^{-1} K$ and b_1, b_2 are the first n^2 and last n^2 components of b. The main idea now is that the computationally dominating part in the computation of the shift-invert operation is the solving of the linear system corresponding to the matrix L, i.e., given $c \in \mathbb{C}^{n^2}$ compute $y \in \mathbb{C}^{n^2}$ such that $Ly = c$. It turns out that if we choose a shift for which $\sigma^{-1} = \overline{\sigma}$ (i.e., $\sigma \in \partial\mathbb{D}$) the operator L is the vectorization of a Lyapunov operator in the following sense. Note that

$$L = \sigma I \otimes A_1 + I \otimes A_0 + A_0 \otimes I + \overline{\sigma} A_1 \otimes I = I \otimes (\sigma A_1 + A_0) + (\overline{\sigma} A_1 + A_0) \otimes I.$$

Suppose $C, Y \in \mathbb{C}^{n \times n}$ are vectorizations of $c, y \in \mathbb{C}^{n^2}$ correspondingly. Then

$$(\overline{\sigma} A_1 + A_0) Y + Y (\overline{\sigma} A_1 + A_0)^* = C. \tag{3.64}$$

This type of matrix equation is often referred to as a Lyapunov equation.

We have just shown that if we can solve (3.64) efficiently, we can improve the efficiency of the shift-invert operation. There are several efficient methods to solve (3.64). In the example below we will use the solver `lyap` bundled with Matlab 7.1.0.183 R14 (which is based on the SLICOT routines `SB03MD, SG03AD, SB04MD`).

The Lyapunov equation (3.64) must be solved several times for the same shift σ but for different B. It may therefore be a good idea to first transform

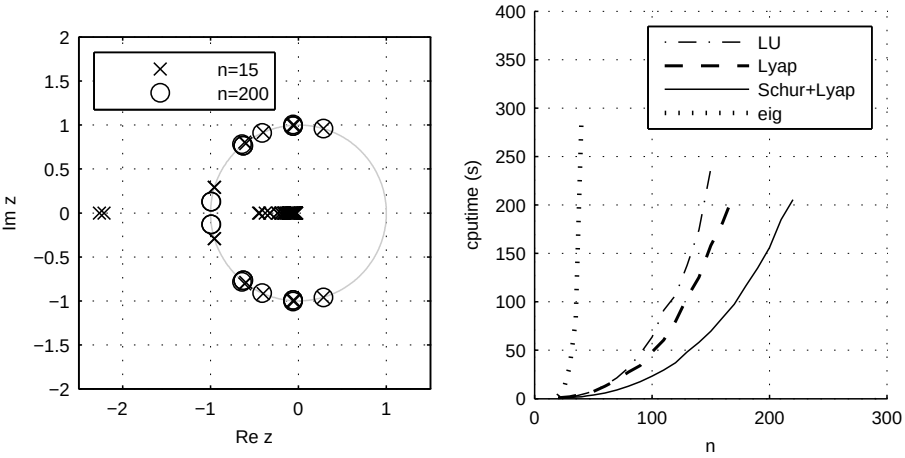
the Lyapunov equation to triangular form with a Schur decomposition of $\overline{\sigma}A_1 + A_0$, as the Lyapunov equation for triangular matrices can be solved with back-substitution. This turns out to be the most efficient approach for the example below.

3.5.2 Example

We now wish to show one example where the improved shift-invert operation allows us to treat larger systems, too large to be treated by directly computing all eigenvalues. Consider the partial delay-differential equation in Example 3.23 with $\tau = \tau_1 = \tau_2$.

We will now use multiple grid-sizes to treat larger systems. In the example we saw that it was possible to treat the problem directly for $n = 14$. From Figure 3.8b, it is clear that this can not be done in practice for system of dimension larger than $n = 40$. It can be seen in Figure 3.8a that the solutions to the quadratic eigenvalue problem are only moderately perturbed when the grid is refined. Hence, it is natural to use the solutions z for the coarse grid $n = 14$ as shifts for a finer grid. Since we are only interested in eigenvalues on the unit circle, we ignore eigenvalues in the coarse grid which have a distance larger than 0.1 from the unit circle. In the Lyapunov exploitation above (i.e. (3.64)) we assumed that the shift is on the unit circle. The shifts are hence forced to be of unit magnitude. We fit the eigenvalue solver `eigs` bundled with Matlab, with different shift-invert operations. The LU-decomposition of (3.63) is a common choice for a shift-invert operation. For this particular example, the LU-decomposition is very sparse (since A_0 and A_1 are tridiagonal matrices) which makes this shift-invert operation perform rather well. However, the two optimizations using `lyap` described above perform better and are expected to have an even larger efficiency improvement in comparison to the LU-decomposition if A_0 and A_1 are full matrices.

It is interesting to note that only 4 eigenvalues z are on the unit circle (closer than 10^{-13}) whereas there are 20 eigenvalues which have a distance to the unit circle less than 0.02.



(a) Eigenvalues of (3.22)

(b) Complexity comparison

Figure 3.8: Computational complexity with structure exploitation

3.6 Multiparameter problems and matrix pencil methods

We mentioned in the review of delay-dependent stability results in Section 3.2 that one class of delay-dependent stability results for DDEs yield an exact way to give conditions in terms of an eigenvalue problem constructed by Kronecker products. These type of methods are referred to as *matrix pencil methods*.

We wish to provide further understanding to these methods, i.e., the main results of the articles [CGN95], [Lou01], [FNC05], [NFC05], [NFC06], [FNC06] and [EOF07], by deriving the essential parts of the methods in a unified way. The parameterizations in Section 3.3 and Section 3.4 also belong to this class of problems.

Eigenvalue problems consisting of Kronecker products also occur in the field for multiparameter eigenvalue problems [Atk72]. See also [HP03] for a presentation of some numerical properties of multiparameter eigenvalue problems. In

particular, multiparameter eigenvalue problems can be rewritten into an equivalent generalized eigenvalue problem which is formed by Kronecker products.

The main result of this section is that the matrix pencil methods are all different determinental forms of a generalized multiparameter eigenvalue with a polynomial term. We refer to this generalization of the multiparameter eigenvalue problem as a *polynomial multiparameter eigenvalue problem*. The determinental forms of the polynomial multiparameter eigenvalue problems are polynomial eigenvalue problems. Note that there are methods for the two-parameter eigenvalue problem which does not explicitly use the determinental form, e.g. the variant Jacobi-Davidson [HKP05].

Multiparameter eigenvalue problems and multivariate polynomials have previously been used in the field of time-delay systems, e.g. [Kam82], [Kam80], [HIT85], [Chi88] and summarized in the book of Niculescu [Nic01a, Section 4.1.2] and the book of Gu, Kharitonov and Chen [GKC03, Section 4.6]. However, these works do not apply theory in the context or derivation of matrix pencil methods.

We start by developing a general theory for the polynomial multiparameter eigenvalue problems in Section 3.6.1. In particular we prove some determinental form of the generalization of the two-parameter eigenvalue problem. The main idea of this section is shown by applying the theory to the single delay DDE in Section 3.6.2. These results are generalized to neutral DDEs in Section 3.6.3 and to commensurate delays in Section 3.6.4.

Parts of the results of this section were produced in collaboration with Michiel E. Hochstenbach.

3.6.1 Polynomial two-parameter eigenvalue problems

Consider the two-parameter eigenvalue problem

$$\begin{cases} A_1 x &= \lambda B_1 x + \mu C_1 x \\ A_2 y &= \lambda B_2 y + \mu C_2 y. \end{cases} \tag{3.65}$$

with $A_k, B_k, C_k \in \mathbb{C}^{n\times n}$, $k = 1, 2$. See the book [Atk72] for a detailed study of multiparameter eigenvalue problems. A generalized eigenvalue problem associ-

ated with (3.65) can be stated in terms of the matrix determinants

$$\begin{aligned} \Delta_0 &= B_1 \otimes C_2 - C_1 \otimes B_2 \\ \Delta_1 &= A_1 \otimes C_2 - C_1 \otimes A_2 \\ \Delta_2 &= B_1 \otimes A_2 - A_1 \otimes B_2 \end{aligned} \tag{3.66}$$

where $\Delta_i \in \mathbb{C}^{n^2 \times n^2}$, $i = 0, 1, 2$. The two-parameter eigenvalue problem (3.65) is expressed as the decoupled generalized eigenvalue problems

$$\Delta_1 z = \lambda \Delta_0 z \tag{3.67a}$$

$$\Delta_2 z = \mu \Delta_0 z, \tag{3.67b}$$

where $z \in \mathbb{C}^{n^2}$. These two formulations are equivalent if Δ_0 is nonsingular. In this work we refer to these two eigenvalue problems as the determinental forms of (3.65). It is important to note that there are two generalized eigenvalue problems corresponding to the two-parameter eigenvalue problem (3.65). We will see in the derivations of the matrix pencil methods that some correspond to (3.67a) and some to (3.67b).

The determinental forms can be generalized to two wider classes of problems. We will make extensive use of the following two lemmas in the derivation of the matrix pencil methods in the following subsections.

Consider the problem of determining $x, y \in \mathbb{C}^n$ and $\lambda, \mu \in \mathbb{C}$ such that

$$\begin{cases} A_1 x &= \lambda B_1 x + \mu C_1 x + \mu\lambda D_1 x \\ A_2 y &= \lambda B_2 y + \mu C_2 y + \mu\lambda D_2 y. \end{cases} \tag{3.68}$$

This is clearly a generalization of the two-parameter eigenvalue problem (3.65). This problem can also be restated into an eigenvalue problem. Unlike the determinental forms of the two-parameter eigenvalue problem, the determinantal forms of the polynomial two-parameter eigenvalue problem (3.68) are quadratic eigenvalue problems.

Lemma 3.38 *If* (λ, μ) *is an eigenvalue of* (3.68) *with corresponding eigenvector* (x, y) *then*

a) λ *is an eigenvalue of the quadratic eigenvalue problem*

$$\begin{aligned} &[(A_1 \otimes C_2 - C_1 \otimes A_2) + \lambda(A_1 \otimes D_2 - D_1 \otimes A_2 + \\ &\quad -B_1 \otimes C_2 + C_1 \otimes B_2) + \lambda^2 (D_1 \otimes B_2 - B_1 \otimes D_2)] \, (x \otimes y) = 0. \end{aligned} \tag{3.69}$$

b) μ is an eigenvalue of the quadratic eigenvalue problem

$$\begin{aligned}&[(A_1 \otimes B_2 - B_1 \otimes A_2) + \mu(A_1 \otimes D_2 - D_1 \otimes A_2 + \\ &-C_1 \otimes B_2 + B_1 \otimes C_2) + \mu^2(D_1 \otimes C_2 - C_1 \otimes D_2)\big]\,(x \otimes y) = 0. \quad (3.70)\end{aligned}$$

Proof: It is sufficient to show a), since the problem (3.68) is symmetric with respect to switching of λ with μ and B_1, B_2 with C_1, C_2.

Equation (3.69) holds because

$$\begin{aligned}&\lambda^2(D_1 \otimes B_2 - B_1 \otimes D_2)(x \otimes y) = \\ &\lambda(D_1 \otimes (A_2 - \mu C_2 - \mu\lambda D_2) - (A_1 - \mu C_1 - \mu\lambda D_1) \otimes D_2)(x \otimes y) = \\ &\lambda(D_1 \otimes (A_2 - \mu C_2) - (A_1 - \mu C_1) \otimes D_2)(x \otimes y) = \\ &(\lambda(D_1 \otimes A_2 - A_1 \otimes D_2) + \lambda\mu(C_1 \otimes D_2 - D_1 \otimes C_2))(x \otimes y) = \\ &(\lambda(D_1 \otimes A_2 - A_1 \otimes D_2) + (C_1 \otimes (A_2 - \lambda B_2) - (A_1 - \lambda B_1) \otimes C_2)(x \otimes y),\end{aligned}$$

where we used that

$$\begin{aligned}&\lambda\mu(C_1 \otimes D_2 - D_1 \otimes C_2)(x \otimes y) = \\ &(C_1 \otimes (A_2 - \lambda B_2 - \mu C_2) - (A_1 - \lambda B_1 - \mu C_1) \otimes C_2)(x \otimes y) = \\ &(C_1 \otimes (A_2 - \lambda B_2) - (A_1 - \lambda B_1) \otimes C_2)(x \otimes y).\end{aligned}$$

□

At the cost of technicalities and longer formulas in the derivation, the lemma partially generalizes to the following class of problems. Find $\lambda, \mu \in \mathbb{C}$, $x, y \in \mathbb{C}^n \backslash \{0\}$ such that

$$\left\{ \begin{array}{ll} A_1 x & = \lambda \sum_{k=0}^m \mu^k B_{1,k} x + \quad \sum_{k=1}^m \mu^k C_{1,k} x \\ A_2 y & = \lambda \sum_{k=0}^m \mu^k B_{2,k} y + \quad \sum_{k=1}^m \mu^k C_{2,k} y \end{array} \right. . \quad (3.71)$$

Note that symmetry with respect to μ and λ is broken. For this generalization of the two-parameter eigenvalue problem, we only find one determinental form. The corresponding problem is a polynomial eigenvalue problem. Note that λ is not occuring in the polynomial eigenvalue problem below.

Lemma 3.39 *If* (λ, μ) *is an eigenvalue of* (3.71) *with eigenvector* (x, y) *then*

$$\Bigg[(A_1 \otimes B_{2,0} - B_{1,0} \otimes A_2) + \sum_{k=1}^{m} \mu^k (A_1 \otimes B_{2,k} - B_{1,k} \otimes A_2 - C_{1,k} \otimes B_{2,0} + B_{1,0} \otimes C_{2,k}) + \sum_{k=1,i=1}^{m} \mu^{k+i} (B_{1,k} \otimes C_{2,i} - C_{1,k} \otimes B_{2,i}) \Bigg] (x \otimes y) = 0.$$

Proof: From (3.71) it follows that

$$\begin{aligned} A_1 x &= \lambda B_1 x + \quad \mu C_1 x + \mu\lambda D_1 x \\ A_2 y &= \lambda B_2 y + \quad \mu C_2 y + \mu\lambda D_2 y, \end{aligned}$$

if we let $B_i = B_{i,0}$, $D_i = \sum_{k=1}^{m} \mu^{k-1} B_{i,k}$ and $C_i = \sum_{k=1}^{m} \mu^{k-1} C_{i,k}$ for $i = 1, 2$. Applying Lemma 3.38 yields that

$$\begin{aligned} 0 = \Bigg[& (A_1 \otimes B_{2,0} - B_{1,0} \otimes A_2) + \\ & + \mu (A_1 \otimes \sum_{k=1}^{m} \mu^{k-1} B_{2,k} - \sum_{k=1}^{m} \mu^{k-1} B_{1,k} \otimes A_2 + \\ & - \sum_{k=1}^{m} \mu^{k-1} C_{1,k} \otimes B_{2,0} + B_{1,0} \otimes \sum_{k=1}^{m} \mu^{k-1} C_{2,k}) + \\ & \mu^2 (\sum_{k=1}^{m} \mu^{k-1} B_{1,k} \otimes \sum_{k=1}^{m} \mu^{k-1} C_{2,k} - \sum_{k=1}^{m} \mu^{k-1} C_{1,k} \otimes \sum_{k=1}^{m} \mu^{k-1} B_{2,k}) \Bigg] (x \otimes y) = \\ = \Bigg[& (A_1 \otimes B_{2,0} - B_{1,0} \otimes A_2) + \\ & + (A_1 \otimes \sum_{k=1}^{m} \mu^{k} B_{2,k} - \sum_{k=1}^{m} \mu^{k} B_{1,k} \otimes A_2 + \\ & - \sum_{k=1}^{m} \mu^{k} C_{1,k} \otimes B_{2,0} + B_{1,0} \otimes \sum_{k=1}^{m} \mu^{k} C_{2,k}) + \\ & (\sum_{k=1}^{m} \mu^{k} B_{1,k} \otimes \sum_{k=1}^{m} \mu^{k} C_{2,k} - \sum_{k=1}^{m} \mu^{k} C_{1,k} \otimes \sum_{k=1}^{m} \mu^{k} B_{2,k}) \Bigg] (x \otimes y), \end{aligned}$$

which completes the proof. □

3.6.2 One single delay

In order to ease the presentation, we start the discussion by treating retarded DDEs with a single delay. We generalize the results to multiple delays and the neutral case in later subsections.

We derive a polynomial two-parameter eigenvalue problem from the characteristic equation for the case that there is a purely imaginary eigenvalue. It turns out that the eigenvalue problems which are the results of the applications of Lemma 3.38 and Lemma 3.39, are the polynomial eigenvalue problems in the works on matrix pencil methods.

Consider

$$B_0\dot{x}(t) = A_0x(t) + A_1x(t-\tau), \tag{3.72}$$

where $A_0, A_1, B_0 \in \mathbb{C}^{n\times n}$. The characteristic equation is

$$B_0sv = (A_0 + A_1e^{-s\tau})v. \tag{3.73}$$

Suppose that there is an imaginary eigenvalue $s = i\omega$. We denote $z = e^{-s\tau}$. Note that for an imaginary eigenvalue $s^* = -s$ and $z^* = z^{-1}$. Hence, we form the conjugate of (3.73)

$$-\overline{B_0}su = (\overline{A_0} + \overline{A_1}z^{-1})u, \tag{3.74}$$

where $u = \overline{v}$. We now rearrange the terms of (3.73) and (3.74) in such a way that we can easily compare it to (3.68). The equations (3.73) and (3.74), yield that

$$\left\{\begin{array}{llll} A_0v & = sB_0v & -zA_1v & \\ \overline{A_1}u & = & -z\overline{A_0}u & -sz\overline{B_0}u \end{array}\right. \tag{3.75}$$

from which we identify that it is a special case of the polynomial two-parameter eigenvalue problem (3.68). Note that there are other ways to construct a polynomial two-parameter eigenvalue problem from the characteristic equation. Instead of using the complex conjugate as a second condition we might just as well take the complex conjugate transpose.

The two quadratic eigenvalue problems in Lemma 3.38 correspond to the two types of elimination done for z in [Lou01] (and mentioned in the review in Theorem 3.9) and for s in [CGN95],[Nic98], [Jar06a].

By applying Lemma 3.38 to (3.75), we find that

$$\left[(A_1 \otimes \overline{A_1}) + s(B_0 \otimes \overline{A_0} - A_0 \otimes \overline{B_0}) + s^2(B_0 \otimes \overline{B_0})\right](v \otimes u) = 0 \tag{3.76}$$

and

$$\left[(B_0 \otimes \overline{A_1}) + z(A_0 \otimes \overline{B_0} + B_0 \otimes \overline{A_0}) + z^2(A_1 \otimes \overline{B_0})\right](v \otimes u) = 0. \tag{3.77}$$

The method of Louisell [Lou01] is constructed for neutral DDEs. For the retarded case, the eigenvalue problem which must be solved in his method corresponds exactly to a linearization of (3.76). More precisely, suppose that all matrices in Theorem 3.9 are real and $B_0 = I$, $B_1 = 0$ (in the notation in Theorem 3.9), then (3.76) reduces to the eigenvalue problem in Theorem 3.9. We discuss the more general case of neutral DDEs in the next subsection.

The derivation of a special case of [CGN95] can be done by considering the complex conjugate transpose of (3.73) (instead of the conjugate). The resulting equations are

$$\begin{cases} A_0 v &= sB_0 v &- zA_1 v \\ A_1^* z &= &- zA_0^* z &- szB_0^* z, \end{cases} \tag{3.78}$$

where z is the left eigenvector of (3.72), i.e., $M(s)^* z = 0$. Lemma 3.38 yields that

$$\left(z^2(A_1 \otimes B_0^*) + z(B_0 \otimes A_0^* + A_0 \otimes B_0^*) + B_0 \otimes A_1^*\right)(x \otimes z) = 0,$$

which is a special case of the eigenvalue problem occurring in [CGN95, Nic98] for the commensurate (but real) case, for *delay-differential algebraic equations* in [NFC06] and for *neutral delay-differential equations* in [FNC06].

We consider these more general cases in the following subsections.

3.6.3 Neutral systems

Consider the neutral DDE

$$B_0\dot{x}(t) + B_1\dot{x}(t-\tau) = A_0 x(t) + A_1 x(t-\tau),$$

where $A_0, A_1, B_0, B_1 \in \mathbb{C}^{n\times n}$. It turns out that Lemma 3.38 can be applied analogously to this more general case and delay-dependent matrix-pencil methods

for neutral DDE can be derived similarly. As in Section 3.6.2, we set $s = i\omega$ and $z = e^{-i\tau\omega}$, and note that the characteristic equation and its complex conjugate can be expressed as

$$\begin{cases} A_0 v & = sB_0 v & -zA_1 v + szB_1 v \\ \overline{A_1} u & = -s\overline{B_1} u & -z\overline{A_0} u - sz\overline{B_0} u. \end{cases} \tag{3.79}$$

After applying Lemma 3.38 we get that

$$\begin{aligned} \big[(-A_0 \otimes \overline{A_0} + A_1 \otimes \overline{A_1}) + s(-A_0 \otimes \overline{B_0} - B_1 \otimes \overline{A_1} + \\ B_0 \otimes \overline{A_0} + A_1 \otimes \overline{B_1}) + s^2(-B_1 \otimes \overline{B_1} + B_0 \otimes \overline{B_0})\big]\,(v \otimes u) = 0, \end{aligned}$$

and after rearrangement of the terms

$$\big((sB_0 - A_0) \otimes (s\overline{B_0} + \overline{A_0}) - (sB_1 - A_1) \otimes (s\overline{B_1} + \overline{A_1})\big)\,(v \otimes u) = 0, \tag{3.80}$$

which is the same general form of the eigenvalue problem as presented by Louisell in [Lou01] and Theorem 3.9, if we assume that $B_0 = I$ and that the matrices are real. Louisell suggests that (3.80) should be solved by solving the generalized eigenvalue problem

$$s \begin{pmatrix} I \otimes I & B_1 \otimes I \\ I \otimes B_1 & I \otimes I \end{pmatrix} x = \begin{pmatrix} A_0 \otimes I & A_1 \otimes I \\ -I \otimes A_1 & -I \otimes A_0 \end{pmatrix} x,$$

motivated by a connection to a differential equation. Here, we note that this is just one possible linearization of (3.80) and any of the linearizations in [MMMM06b, MMMM06a] could be used.

3.6.4 Commensurate delays

Consider the DDE with commensurate delays,

$$B_0 \dot{x}(t) = \sum_{k=0}^{m} A_k x(t - hk),$$

which has the characteristic equation

$$\left(\sum_{k=0}^{m} e^{-hks} A_k - sI\right) v = 0.$$

As in the previous section we substitute $s = i\omega$ and $z = e^{-i\tau\omega}$ and consider the conjugate of the characteristic equation. After rearrangement of the terms and sums we have

$$\begin{cases} -\overline{A_m}u = & sz^m\overline{B_0}u + \sum_{k=1}^m z^k\overline{A_{m-k}}u \\ A_0 v = & sB_0 v - \sum_{k=1}^m z^k A_k v. \end{cases} \tag{3.81}$$

This is the polynomial two-parameter eigenvalue problem in (3.71) with $A_1 = -A_m$, $B_{1,m} = B_0$, $B_{1,k} = 0$, $k = 0, \dots m-1$, $C_{1,k} = A_{m-k}$, $k = 1, \dots, m$, $A_2 = A_0$, $B_{2,0} = B_0$, $B_{2,k} = 0$, $k = 1, \dots m$, $C_{2,k} = -A_k$, $k = 1, \dots, m$.

Lemma 3.39 and several manipulations of the sums yield

$$\begin{aligned} 0 = \Bigg[-A_m \otimes B_0 + z^m(-B_0 \otimes A_0) + \sum_{k=1}^m z^k(-A_{m-k} \otimes B_0) + \\ \sum_{i=1}^m z^{m+i}(-B_0 \otimes A_i) \Bigg] (v \otimes u) = \\ \left[-\sum_{k=0}^m z^{m-k}(A_k \otimes B_0) - \sum_{i=0}^m z^{m+i}(B_0 \otimes A_i) \right] (v \otimes u). \end{aligned} \tag{3.82}$$

Note that (3.82) is the same polynomial eigenvalue problem we derived in Section 3.3.3, i.e., (3.27).

Similarly, if we consider the conjugate transpose instead of the transpose, Lemma 3.39 gives the polynomial eigenvalue problem in [CGN95, Theorem 3.1].

Finally, our most general result is for neutral commensurate DDEs. We show that the eigenvalue problem in [FNC06] is also a determinental form of a polynomial two-parameter eigenvalue problem, with a similar analysis as for the previous cases. The price for the wider generality is payed with technicalities and larger expressions. Consider the characteristic equation of the neutral commensurate DDE

$$\sum_{k=0}^m B_k \dot{x}(t - hk) = \sum_{k=0}^m A_k x(t - hk),$$

i.e.,

$$\left(A_0 + \sum_{k=0}^m -sz^k B_k + \sum_{k=1}^m z^k A_k \right) v = 0. \tag{3.83}$$

The complex conjugate transpose is

$$\left(z^m A_0^* + \sum_{k=0}^{m} s z^{m-k} B_k^* + \sum_{k=1}^{m} z^{m-k} A_k^* \right) u = 0. \tag{3.84}$$

We can now combine (3.83) and (3.84) into a polynomial two-parameter eigenvalue problem

$$\left\{ \begin{array}{rcl} A_0 v & = & s \left(\sum_{k=0}^{m} z^k B_k \right) v - \sum_{k=1}^{m} z^k A_k v \\ -A_m^* u & = & s \sum_{k=0}^{m} z^k B_{m-k}^* u + \sum_{k=1}^{m} z^k A_{m-k}^* u. \end{array} \right. \tag{3.85}$$

This corresponds to (3.71) with $A_1 = A_0$, $B_{1,k} = B_k$, $k = 0, \ldots m$, $C_{1,k} = -A_k$, $k = 1, \ldots, m$, $A_2 = -A_m^*$, $B_{2,k} = B_{m-k}^*$, $k = 1, \ldots m$, $C_{2,k} = A_{m-k}^*$, $k = 1, \ldots, m$. Lemma 3.39 yields

$$\begin{bmatrix} (A_0 \otimes B_{m-0}^* + B_0 \otimes A_m^*) + \\ \sum_{k=1}^{m} z^k (A_0 \otimes B_{m-k}^* + B_k \otimes A_m^* + A_k \otimes B_{m-0}^* + B_0 \otimes A_{m-k}^*) + \\ \sum_{k=1,i=1}^{m} z^{k+i} (B_k \otimes A_{m-i}^* + A_k \otimes B_{m-i}^*) \end{bmatrix} (v \otimes u) = 0. \tag{3.86}$$

The coefficients of (3.86) can now be compared to the coefficients of the eigenvalue problem in [FNC06, Theorem 2]. With some effort it can be verified that the matrix coefficients Q_k in [FNC06, Theorem 2] are the matrix coefficients in polynomial eigenvalue problem (3.86).

3.7 NP-hardness issues

The result on NP-hardness by Toker and Özbai [TÖ96] is commonly cited in works related to stability margin for time-delay systems. We now discuss how these results relate to the proposed methods in Section 3.3 and Section 3.4. Large parts of the discussion are the results of a collaboration with Henrik Bäärnhielm.

With the discussion we wish to clarify a common misinterpretation of these results and stress that the NP-hardness results do not imply that the problem

considered in this chapter is difficult in the sense normally relevant in numerical analysis. Moreover, the methods proposed in previous sections neither contradict nor verify the NP-hardness results. We will discuss the following reasons:

- The problem we consider in this chapter and the problem in [TÖ96] are not the same.
- The method we suggest involves the computation of eigenvalues and can not deliver exact results on a Turing machine.
- A problem considered difficult in computer science is not necessarily difficult in the context of numerical analysis.

We will also discuss variants of the results by Toker and Özbai attempted in [GKC03].

In theoretical computer science, the class of problems referred to as NP-hard corresponds to some problems considered particularly hard to solve, but computationally easy to verify (problems at least as difficult as nondeterministic polynomial-time). See [BSS98] or any undergraduate text-book in complexity theory for a precise definition. It is generally believed but not proven that NP-hard problems do not allow a polynomial-time solution scheme. The theorem below by Toker and Özbai [TÖ96] states, in rough terms, that given rational matrices and rational delays defining a rectangle in delay-space, the problem of determining whether all points in the delay-space rectangle are stable is NP-hard.

Theorem 3.40 ([TÖ96]) *The following robust stability problem is NP-Hard: Given $A_0, \ldots, A_m$ where $A_i \in \mathbb{Q}^{n\times n}$ and nonnegative numbers h_i^-, $h_i^+ \in \mathbb{Q}$, $i = 1, \ldots, m$, is the system*

$$\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{m} A_k x(t - h_k)$$

asymptotically stable for all $h_i \in [h_i^-, h_i^+]$, $i = 1, \ldots, m$?

At first sight, the result above may seem to suggest that the problem we are considering in this chapter is difficult. This interpretation is not correct in the context of numerical analysis. It does not imply that the problem is difficult from a numerical point of view.

Clearly, the problem is formulated in a different way in comparison to what is considered in the previous sections. That is, the determination of the stability of any delay-space rectangle is not the same as parameterizing the delay-space points which are stable and also not the same as the problem of determining for which points there are imaginary eigenvalues. The problems are not necessarily computationally equivalent which is one simple motivation why the NP-hardness result can not be applied to the problem considered in this chapter. For the sake of argument we will ignore this difference of formulation in the discussion that follows.

Typically in computer science (as is the case in Theorem 3.40) it is assumed that the hardware upon which the problem is NP-hard is a Turing machine.

Thus, another motivation why Theorem 3.40 has no implications on our method is the fact that our method presented in this chapter can not deliver exact results on a Turing-machine.

One step of the method we propose involves finding the unit-magnitude eigenvalues of a matrix, which is (theoretically but not numerically) equivalent to finding roots of a polynomial. It follows from the fundamental theorem of Galois-theory (e.g. [BB90]) that there are real roots of polynomials which are not representable by the elementary operations and hence in particular not computable with a Turing-machine. That is, even though an eigenvalue problem is considered to be a tractable problem in numerical analysis, it is not solvable on a Turing-machine. Since the polynomial root-finding problem belongs to an unsolvable class of problems, our method is not implementable exactly on a Turing machine. Moreover, for multiple delays the delay-space is parameterized and to determine if a given rectangle is stable from the parameterization is not necessarily easy (polynomial-time).

We believe that an important reason why the complexity results are often incorrectly interpreted stems from the fact that the meaning of *difficult* (problem) is context dependent. In computer science, problems for which there is a polynomial-time algorithm delivering an exact answer, are typically considered easy. Whereas in numerical analysis we are satisfied with an answer which is correct up to machine precision. But, more importantly, in numerical analysis, polynomial time is normally not enough to classify an algorithm as efficient (and the problem as easy to solve). In fact, in numerical analysis and specifically in numerical linear algebra, the goal is often to be optimal, i.e., to have a compu-

tational effort proportional to the size of the input data.

Consider the problem of computing 2^x where $x \in \mathbb{N}$ is the input. We call size in bits of the input $n := \text{bits}(x) = \log_2(x)$. The size of the output is $\text{bits}(2^x) = \log_2(2^x) = x = 2^n$. Since all of the bits in the output can not be touched in polynomial time, the problem of computing 2^x can never be computed in polynomial time on a Turing machine and is in a sense considered to be a difficult problem in computer science. The corresponding problem in numerical analysis is clearly not considered difficult. In fact, exponentiation is typically seen as a fundamental operation provided by the processor, i.e., it can be computed to machine precision in constant time. With this simple example we have shown that what in computer science is considered difficult may not necessarily be difficult in the context of numerical analysis.

Note that our arguments hold in general; if a problem is considered difficult in computer science it is not necessarily so in numerical analysis, nor does an easy problem (polynomial-time) in computer science imply that is easy from a numerical point of view. In particular, the many NP-hardness results in control theory reviewed by Blondel and Tsitsiklis in [BT00] have very little impact on the difficulty of the corresponding problems from a numerical point of view. It seems that the authors were aware of this, telling from the statement "However this [NP-hardness] is not a reason enough for declaring the problem intractable and refraining from further research".

Another slightly different variant of Theorem 3.40 is presented by Gu, Kharitonov and Chen in [GKC03, Chapter 3.4] where similar NP-hardness results are presented for real input data. The character of complexity theory changes considerably if one allows the underlying hardware to handle (exact) real arithmetic which is here implicitly assumed (see [BSS98]). In particular, it is not clear as to whether the Knapsack-problem used in the proof by Gu et al. is indeed NP-hard on a computer which can handle exact arithmetic. This would indicate that the proof in [GKC03] requires some clarification.

Finally, it is relevant to know if a problem can be approximated in polynomial time. One such class of problems (in computer science) is referred to as FPTAS. Such problems allow a *fully polynomial-time approximation scheme* (e.g. [ACG+99]), i.e., given an accuracy ϵ the problem can be solved to that accuracy in polynomial time. It turns out (as seen in [IK75]) that the Knapsack-problem is FPTAS. Since the proof of Theorem 3.40 involves the reduction to the Knapsack-

problem, this would indicate that the corresponding approximate problem could allow a polynomial-time approximation. However, still, determining if a problem is FPTAS does not reveal if a problem is easy or difficult in the context of numerical analysis. In particular, if the problem in Theorem 3.40 is FPTAS, then that does not necessarily mean that the problem is easy in the context of numerical analysis.

Chapter 4

Perturbation of nonlinear eigenproblems

In a survey paper on eigenvalue problems in year 2000, two experts in the field of numerical linear algebra wrote:

> Perturbation theorems play a very essential role in computational processes for eigenproblems.
>
> *Gene Golub and Henk van der Vorst [GvdV00]*

The survey paper is mostly on linear eigenvalue problems. It is however to expect that the role of perturbation results for *nonlinear eigenvalue problems* is, or will be, equally important. In this chapter we will present some notes on how some perturbation results can be generalized to nonlinear eigenvalue problems and how we can use them to determine convergence properties of some numerical methods.

In rough terms, the *eigenvalue perturbation* is described as the qualitative analysis of the behavior of an eigenvalue s of a matrix $A \in \mathbb{C}^{n\times n}$ when the matrix is perturbed. That is, the analysis of the relation between $\tilde{s} \in \sigma(\tilde{A})$ and $s \in \sigma(A)$ when some relation between A and $\tilde{A}$ is given, e.g. in terms of bounds of the norm $\|A - \tilde{A}\| \leq \delta$.

We will distinguish between two types of perturbation results. The behavior of each individual eigenvalue when $\delta \to 0$ is described by the *sensitivity* or *condition*

number of the eigenvalue, which is a local property of individual eigenvalues. We will also present some non-local results where we bound the maximum movement of all eigenvalues in some region. Some sensitivity analysis will be presented in Section 4.3 and some nonlocal results based on a famous perturbation theorem for linear eigenvalue problems (the Bauer-Fike theorem) will be presented in Section 4.4.

One of our goals is to generalize perturbation results for linear eigenvalue problems to nonlinear eigenvalue problems. In order to construct true generalizations to nonlinear eigenvalue problems, we will not use the same form of the nonlinear eigenvalue problem we used in Chapter 2. In Chapter 2 the nonlinear eigenvalue problem was to determine $s \in \mathbb{C}$ such that a parameter dependent matrix M is singular, that is,

$$M(s)v = 0, v \in \mathbb{C}^n \backslash \{0\}. \tag{4.1}$$

Instead we will use a fixed point form of a nonlinear eigenvalue problem. The nonlinear eigenvalue problem in (set-valued) fixed point form is the problem of determining $s \in \mathbb{C}$ such that the parameter dependent matrix G has an eigenvalue s, i.e., $s \in \sigma(G(s))$ or equivalently

$$sv = G(s)v,\ v \neq 0. \tag{4.2}$$

Some of the many ways to transform (4.1) to (4.2) are discussed in Section 4.2. We will exclusively use the fixed point form (4.2) in this chapter because,

1. it allows us to easier state generalizations of perturbation results for the linear case, because the linear case is $G(s) = A$ and a special case of (4.2);
2. the relation $s \in \sigma(G(s))$ is a (so-called) set-valued fixed point problem known in the field of fixed point theory (see e.g. [KB01]);
3. some numerical methods for nonlinear eigenvalue problems can be written as $s_{k+1} \in \sigma(G(s_k))$.

In particular, we will state and prove a sensitivity formula in terms of left and right eigenvectors in Section 4.3 and use some results from fixed point theory to generalize the Bauer-Fike theorem to a class of nonlinear eigenvalue problems in Section 4.4. Moreover, we will see how one can define the convergence order of a fixed point iteration $s_{k+1} \in \sigma(G(s_k))$ in terms of right and left eigenvectors.

4.1 Notes on current literature

We mention some standard references in perturbation theory. The book by Stewart and Sun [SS90] contains many perturbation results for matrices, often expressed in terms of matrix norms. The book by Kato [Kat95] and Baumgärtel's monograph [Bau85] serve as comprehensive studies of perturbation theory including (more generally) linear operators. The book of Trefethen and Embree [TE05] contain many perturbation results related to non-normal eigenvalue problems.

Even though perturbation theory has received a lot of attention for a long period of time, the amount of literature directly applicable to the nonlinear eigenvalue problem we are considering here is limited. We will now discuss some of the more recent results related to the nonlinear eigenvalue problem.

The *pseudospectra* (or sometimes *spectral value set*) are popular when it comes to visualization and analysis of perturbations. The most popular overview of pseudospectra is the commonly cited book of Trefethen and Embree [TE05]. The ε-pseudospectrum of matrix A is the set of values $s \in \mathbb{C}$ for which s is an eigenvalue of a perturbed matrix $\tilde{A}$ such that $\|\tilde{A} - A\| \leq \varepsilon$, i.e.,

$$\sigma_\varepsilon(A) := \{s \in \mathbb{C} : \exists E \in \mathbb{R}^{n\times n}, \|E\| \leq \varepsilon \text{ such that } s \in \sigma(A + E)\}. \tag{4.3}$$

Several attempts have been made to generalize results on pseudospectra to nonlinear eigenvalue problems. The first appearance of pseudospectra for nonlinear eigenvalue problems directly related to the type of problems here, was probably the article of Cullum and Ruehli [CR01]. Cullum and Ruehli defined the nonlinear pseudospectra for the problem $\det(T(s)) = 0$ as

$$\Phi_\varepsilon^{nl}(T) := \{s \in \mathbb{C} : \kappa_2(T(s)) = \|T(s)\|_2\|(T(s))^{-1}\|_2 \geq \varepsilon^{-1}\}. \tag{4.4}$$

They presented a numerical procedure to plot $\Phi_\varepsilon^{nl}(T)$ in a specified region and applied it to determine the nonlinear pseudospectra corresponding to the poles and zeros of a delay system from a model of a *partial element equivalent circuits*. Note that the definition (4.4) is not a generalization of the usual definition of pseudospectra of a matrix as it does not reduce to (4.3) if $T = A - sI$. This can be seen as follows: We know, from e.g. [TE05, Theorem 2.1], that the definition (4.3) is equivalent to $\sigma_\varepsilon(A) = \{s \in \mathbb{C} : \|(A - sI)^{-1}\| > \varepsilon^{-1}\}$. This should be compared to the definition of Cullum and Ruehli applied to $T(s) = A - sI$, i.e., $\Phi_\varepsilon^{nl}(A - sI) = \{s \in \mathbb{C} : \|A - sI\|\|(A - sI)^{-1}\| \geq \varepsilon^{-1}\}$. We note that Φ_ε^{nl} has the additional (nonlinear) factor $\|A - sI\|$.

Several properties of the usual definition of pseudospectra are however maintained in the modified extension (4.4). The eigenvalues are always contained in the pseudospectrum, i.e., for all eigenvalues s, i.e., s such that $\det(T(s)) = 0$, $s \in \Phi_\varepsilon^{nl}(T)$ for any $\varepsilon > 0$ (since the condition number of a singular matrix is defined as ∞). Moreover, $\Phi_{\varepsilon_1}^{nl}(T) \subset \Phi_{\varepsilon_2}^{nl}(T)$ when $\varepsilon_1 < \varepsilon_2$.

Michiels, Green, Wagenknecht and Niculescu have published several results on pseudospectra for nonlinear eigenvalue problems and in particular the delay eigenvalue problem [GW06], [WMG08] and [MGWN06]. Their definition of pseudospectra for nonlinear eigenvalue problems is a true generalization of (4.3). In the most general form (e.g. [MGWN06]), they consider problems which consist of a sum of products of matrices and analytic functions. In particular, they consider perturbations of $T(s) = \sum_{k=0}^{m} A_k f_k(s)$, where $A_k \in \mathbb{C}^{n\times n}$, $f : \mathbb{C} \to \mathbb{C}$, and define the pseudospectra as

$$\begin{aligned}\sigma_\varepsilon(T) := \quad & \{s \in \mathbb{C} : \det(T(s) - \Delta T(s)) = 0 \text{ where} \\ & \Delta T(s) = \sum_{k=0}^{m} f_k(s)\delta A_k \text{ and } \delta A_k \in \mathbb{C}^{n\times n},\ \|\delta A_k\| < \varepsilon\alpha_k\}. \quad (4.5)\end{aligned}$$

This is a direct generalization of the definition of pseudospectra for polynomial eigenvalue problems in [TH01]. Note that the uncertainty $\Delta T(s)$ is a perturbation of the coefficient matrices, i.e., the functions f_k in T and ΔT are not changed. Generalizations of the definition of pseudospectra as well as several equivalence relations and relations to stability radii were presented for the polynomial case, i.e., $f_k(s) = s^k$, in [TH01]. A computational procedure was also presented. Many of these results, including the computational procedure, were generalized to arbitrary (analytic) f_k in [MGWN06] and [WMG08].

Green and Wagenknecht compared definition (4.5) with an alternative (not equivalent) definition for DDEs in [GW06]. We saw in Chapter 2 that the eigenvalues of a DDE are the eigenvalues of a corresponding linear infinite-dimensional system (i.e., infinitesimal generator). Since pseudospectra for linear operators can be defined analogous to (4.3), the pseudospectra of the infinitesimal generator is an alternative definition of the pseudospectra of a DDE. In the numerical method, the discretization infinitesimal generator is refined until the change of the pseudospectra (of the corresponding matrix) in the relevant region is less than some tolerance. In [GW06], the infinitesimal generator is discretized using a Chebyshev method.

Structured (eigenvalue) perturbation is a description of the change of the eigenvalues when the perturbation is such that the perturbed problem is restricted to some subset of the general problem. Typically, the subset is such that some properties of either the solution or the problem are preserved under the perturbation. For instance, if an application for physical reasons generates a real matrix, then a perturbation analysis with a complex perturbation matrix is likely to yield pessimistic bounds on the eigenvalues in comparison to an analysis where the perturbed matrix is also real. Other examples of relevant structured problems are symmetric, skew-symmetric, persymmetric, orthogonal, symplectic, nonnegative matrices, matrix products, etc. See [FK06] for more types of structures.

The pseudospectra definition (4.5) maintains the functions whereas the matrices are perturbed. Therefore, $\sigma_\varepsilon(T)$ in (4.5) is sometimes referred to as a structured pseudospectrum. See also [TE05, Chapter 50] for structured pseudospectra.

Various results for linear eigenvalue problems have been generalized to specific structured eigenvalue problems. This holds in particular for the polynomial eigenvalue problem. Without being exhaustive we will mention some results. Apart from the already mentioned result [TH01], a numerical method based on a predictor-correction tracing was suggested in [LP05]. The Bauer-Fike theorem (see Section 4.4.1), which is a popular way to bound the spectra using the condition number has been generalized to polynomial eigenvalue problems in [Chu03].

The polynomial eigenvalue problem is a very special nonlinear eigenvalue problem since it allows a transformation to a linear eigenvalue problem. The most common (so-called) linearization is the *companion linearization*, which we have used several times in other chapters. This simple form is unfortunately unsuitable for many cases as the linearization does not preserve structure of the underlying matrices. Different linearization should be used for different matrix structures. A complete characterization of the possible linearizations was given by Mackey, Mackey, Mehl and Mehrmann [MMMM06b]. The same authors discussed different structured linearizations in [MMMM06a].

The pseudospectra of the companion form (mostly for scalar problems) was investigated in [TT94], where the importance of balancing was geometrically established. That is, the similarity transformation with the diagonal matrix may change the conditioning (and pseudospectra) of the eigenvalue problem a lot. See also the method for optimal scaling for polynomial eigenvalue problems in

[Bet07b].

Finally, we have results on structured condition numbers, i.e., the modulus of the sensitivity of individual eigenvalues under structured perturbations. In [KKT06], computable expressions for many different structures are given. It is interesting that for some structured (normwise) perturbations the structured and unstructured condition number is the same [Rum06].

4.2 The fixed point form and a similarity transformation

The nonlinear eigenvalue problem is often written as the problem of finding $s \in \mathbb{C}$ such that $M(s)$ is singular, i.e.,

$$M(s)v = 0, v \in \mathbb{C}^n \backslash \{0\}. \tag{4.6}$$

For the scalar case ($n = 1$) this is a root-finding problem. Clearly, for any M we can rewrite (4.6) into a fixed point form

$$sv = G(s)v,\ v \in \mathbb{C}^n \backslash \{0\} \tag{4.7}$$

in a number of ways. That is, instead of searching for $s \in \mathbb{C}$ such that $M(s)$ is singular we search for $s \in \mathbb{C}$ such that s is an eigenvalue of $G(s)$, i.e., $s \in \sigma(G(s))$. We will denote the solutions of (4.7) with $\sigma(G)$, that is $\sigma(G) := \{s \in \mathbb{C} : s \in \sigma(G(s))\}$. Note that this generalizes the definition of the spectrum of a matrix A, $\sigma(A)$ where A is constant. The fixed point form (4.7) is often used in literature on delay eigenvalue problems, where $G(s) = \sum_{k=0}^{m} A_k e^{-\tau_k s}$.

In this chapter we will use the fixed point form (4.7) because it allows direct generalizations of perturbation results for eigenvalue problems and several methods for nonlinear eigenvalue problems can be written as fixed point iterations $s_{k+1} \in \sigma(G(s_k))$. Moreover, in Section 4.4 we will use that (4.7) is a special case of a *set-valued fixed point problem.*

A similarity transformation

Similar to scalar fixed point problems, the set-valued nonlinear eigenvalue problem (4.7) can be written in many mathematically equivalent ways. Different

mathematically equivalent forms may have very different numerical properties. Apart from the obvious similarity transformation, $\sigma(G(s))= \sigma(Q(s)^{-1}G(s)Q(s))$ for any invertible $Q(s) \in \mathbb{C}^{n\times n}$, we have the following more general class of equivalent nonlinear eigenvalue problems.

Lemma 4.1 (Similarity transformation) *Let $G : \mathbb{C} \to \mathbb{C}^{n\times n}$ be a parameter dependent matrix (corresponding to a nonlinear eigenvalue problem). For any parameter dependent matrices A, B such that $1 \notin \sigma(A(s))$ and $B(s)$ nonsingular for all $s \in \mathbb{C}$, the nonlinear eigenvalue problems corresponding to G and H share solutions, i.e., $\sigma(G) = \sigma(H)$ where*

$$H(s) := (I - A(s))^{-1}(B(s)G(s)B(s)^{-1} - sA(s))). \tag{4.8}$$

Moreover, the eigenvectors u, v are related by

$$u = B(s)v$$

where u, v are eigenvectors of $H(s)$ and $G(s)$ correspondingly.

Proof: Since $B(s)$ is invertible $sv = G(s)v \Leftrightarrow su = B(s)G(s)B(s)^{-1}u$ where $u = B(s)v$. Let $F(s) := B(s)G(s)B(s)^{-1}$ then

$$su = F(s)u \Leftrightarrow s(I - A(s))u = (F(s) - sA(s))u.$$

Since $I - A(s)$ is invertible for all $s \in \mathbb{C}$ by assumption,

$$su = (I - A(s))^{-1}(F(s) - sA(s))u = H(s)u.$$

□

Later, we will make special use of the special case that $B(s) = I$, i.e., $\sigma(G) = \sigma(H)$, where

$$H(s) = (I - A(s))^{-1}(G(s) - sA(s))). \tag{4.9}$$

Note that the lemma above (Lemma 4.1) is a true generalization of the similarity transformation for matrices. If G is constant, i.e., $G(s) = C$, then we the lemma reduces to the usual similarity transformation if we set $A(s) = 0$ and $B(s) = B_0$.

This similarity transformation is not a unique property of the fixed point form. For instance, (4.9) can be interpreted in the context of the matrix singular

form (4.6) as the identity $M(s) = -sI + A(s)^{-1}(M(s) + sA(s))$. Many results in this chapter can be stated as equivalent results for the matrix singular form (4.6). We will stick to the fixed point form and discuss the interpretations in the matrix singular form only where it is illustrative.

In the later sections we will choose $A(s) = G'(s)$, $B(s) = I$ and see that the corresponding fixed point iteration is a form of Newton's method. Other methods for nonlinear eigenvalue problems can be interpreted as fixed point iterations with similarity transformations.

We have introduced the similarity transformation in order to make it easier to interpret some numerical methods for nonlinear eigenvalue problems. Note that some nonlinear eigenvalue problems can be analyzed analytically with the transformation. In the following example we show that Lemma 4.1 can be used to transform the problem to a triangular form where the classical transformation $\sigma(B^{-1}G(s)B) = \sigma(G(s))$ fails, since the matrices are not simultaneously triangularizable.

Example 4.2 *Consider the nonlinear eigenvalue problem*

$$G(s) = \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} f(s).$$

This problem can be explicitly treated by forming the determinant. Here we will demonstrate the use of Lemma 4.1 by solving it using the similarity transformation. Note that for $\alpha \neq 0$ and $f(s) \neq 0$ the matrices in $G(s)$ can not be transformed to triangular form simultaneously using the classical similarity transformation $B^{-1}G(s)B(s)$. If we assume that $s \neq 0$, we can apply (4.9) *with* $A(s) = \begin{pmatrix} 0 & f(s)/s \\ 0 & 0 \end{pmatrix}$ *and transform G to a triangular nonlinear eigenvalue problem H. Here*

$$H(s) = \begin{pmatrix} 1 & -\frac{1}{s}f(s) \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix} = \begin{pmatrix} \frac{\alpha f(s)}{s} & 0 \\ \alpha & 0 \end{pmatrix}.$$

We can characterize all solutions (apart from possibly 0) by $\sigma(G)\backslash\{0\} = \sigma(H)\backslash\{0\} = \{s \in \mathbb{C} : s^2 = \alpha f(s)\}\backslash\{0\}$. If we pick $f(s) = e^{-s}$ the problem reduces to the example for the Lambert W in Section 2.2.1.

4.3 Local perturbation and convergence

4.3.1 Local perturbation and sensitivity analysis

The problem we now consider is, given an eigenvalue, bound or describe the change of this eigenvalue when the problem is perturbed. This is a local property in the sense that we suppose that one eigenvalue is given and that we try to find expressions of the perturbed eigenvalue using the original one. Sensitivity is one such local property.

For linear eigenvalue problems the sensitivity of an eigenvalue is defined as follows. The individual eigenvalues of the matrix $A(h) \in \mathbb{C}^{n\times n}$ (dependending on parameter h) move continuously with respect to changes in h (e.g. [HP05, Corollary 4.2.1]). The union of the spectrum of $A(h)$ for a h in some interval is the union of n continuous functions. Suppose these eigenvalue paths are differentiable at some point h, then the sensitivity of the eigenvalue is the derivative of the corresponding eigenvalue path. For linear eigenvalue problems, the sensitivity of eigenvalue $s \in \sigma(A(h))$ is given (in e.g. [Lan64, Theorem 5] or [HP05, Proposition 4.2.12]) by

$$s'_h(h) = \frac{u^* A'(h) v}{u^* v} \tag{4.10}$$

where u, v are the left and right eigenvectors correspondingly. This sensitivity quotient (or *Rayleigh quotient*) is derived in a couple of different ways in the literature. For instance, in [SS90, Theorem IV.2.3] the formula follows from *Gerschgorin's theorem*, Lancaster [Lan64] uses a *Taylor series* whereas the very simple proof in [HP05, Proposition 4.2.12] is based on implicit differentiation of the eigenvalues and the eigenvectors.

We wish to generalize this sensitivity formula. But first we note some consequences of two types of mean-value theorems.

Lemma 4.3 (Vector-valued mean value theorem) *Suppose f is a continuous function of $[a, b]$ into $\mathbb{R}^k$ and f is differentiable in (a, b). Then there exists $\xi \in (a, b)$ such that*

$$\|f(b) - f(a)\| \leq (b - a)\|f'(\xi)\|.$$

Proof: See [Rud76, Theorem 5.19]. □

Note that the lemma holds in particular for functions with a complex range, i.e., $f : \mathbb{R} \to \mathbb{C}$ (since the real and imaginary parts of $\mathbb{C}$ can be seen as $\mathbb{R}^2$). We use this to derive the following bound of the change of the eigenvalue.

For the upper bound, we need a different mean value theorem.

Theorem 4.4 (Sensitivity) *Suppose $sv^* = v^*G(s)$, $zu = H(z)u$ and $v^*(G(z) - H(z))u \neq 0$. If $G(s)$ is differentiable, then,*

$$|s - z| \geq \frac{|v^*(G(z) - H(z))u|}{|v^*(I - G'(\xi))u|} \tag{4.11}$$

for some $\xi \in l(s, z)$, where $l(a, b)$ denotes the line-segment between a and b.

Proof: Note that

$$|v^*(G(z) - H(z))u| = |v^*(sI - G(s))u - v^*(zI - G(z))u|.$$

Let $\varphi(\theta) = s\theta + z(1 - \theta)$, i.e., a line going through s and z. We define $f(\theta) = v^*(\varphi(\theta)I - G(\varphi(\theta)))u$. Note that by the chain-rule $f'(\theta) = v^*(I - G'(\varphi(\theta)))u(s - z)$. It follows from Lemma 4.3 that

$$|v^*(sI - G(s))u - v^*(zI - G(z))u| = |f(1) - f(0)| \leq |f'(x)| = \\ |v^*(I - G'(\varphi(x)))u||(s - z)|,$$

for some $x \in (0, 1)$. Hence,

$$|s - z| \geq \frac{|v^*(G(z) - H(z))u|}{|v^*(I - G'(\varphi(x)))u|}.$$

□

This local perturbation result gives a lower bound on the movement of the individual eigenvalue s. Note that the range of applicability of Theorem 4.4 is limited, as it does not yield any information if $H \to G$.

Theorem 4.5 (Complex mean-value theorem [EJ92]) *Suppose Ω is an open convex set in $\mathbb{C}$, suppose f is a holomorphic function $f : \Omega \to \mathbb{C}$, and suppose a, b are distinct points in Ω. Then there exist points u, v on $l(a, b)$ such that*

$$\frac{f(b) - f(a)}{b - a} = \text{Re } (f'(u)) + i\text{Im } (f'(v)),$$

where $l(a, b)$ denotes the line-segment between a and b.

Proof: See [EJ92]. □

Theorem 4.6 *Suppose $sv^* = v^*G(s)$, $zu = H(z)u$ and $v^*(G(z) - H(z))u \neq 0$, and suppose $G(s)$ is differentiable. Moreover, let $q(\zeta) := v^*(I - G'(\zeta))u$. Then,*

$$|s - z| \leq \frac{|v^*(G(z) - H(z))u|}{\left|\text{Re } (q(\zeta_1)\overline{q(\zeta_2)})\right|} \left\| \begin{pmatrix} \text{Re } q(\zeta_2) & \text{Im } q(\zeta_1) \\ -\text{Im } q(\zeta_2) & \text{Re } q(\zeta_1) \end{pmatrix} \right\|_2 \tag{4.12}$$

$$= |v^*(G(z) - H(z))u| \left(\sigma_{\min} \begin{pmatrix} \text{Re } q(\zeta_1) & -\text{Im } q(\zeta_1) \\ \text{Im } q(\zeta_2) & \text{Re } q(\zeta_2) \end{pmatrix} \right)^{-1} \tag{4.13}$$

for some $\zeta_1, \zeta_2 \in l(s, z)$.

Proof: Let $f(\theta) := v^* \left(\varphi(\theta)I - G(\varphi(\theta))\right) u$ and $\varphi(\theta) := s\theta - z(1 - \theta)$, i.e., as in the proof of Theorem 4.4. Then $f(1) - f(0) = v^*(G(z) - H(z))u =: \alpha$ and $f'(\theta) = v^*(I - G'(\varphi(\theta)))u(s - z)$, i.e., $q(\varphi(\theta))(s - z) = f'(\theta)$.

Now note that from the complex mean-value theorem (Theorem 4.5), there are $\theta_1, \theta_2 \in l(0, 1)$ such that

$$\begin{aligned} \text{Re } \alpha &= \text{Re } f'(\theta_1) = \text{Re } (q(\varphi(\theta_1))(s - z)) = \text{Re } q_1 \text{Re } (s - z) - \text{Im } q_1 \text{Im } (s - z) \\ \text{Im } \alpha &= \text{Im } f'(\theta_2) = \text{Im } (q(\varphi(\theta_2))(s - z)) = \text{Im } q_2 \text{Re } (s - z) + \text{Re } q_2 \text{Im } (s - z), \end{aligned}$$

where we let $q_1 = q(\varphi(\theta_1)$ and $q_2 = q(\varphi(\theta_2)$ for notational convenience. This is, in matrix form

$$\begin{pmatrix} \text{Re } \alpha \\ \text{Im } \alpha \end{pmatrix} = A \begin{pmatrix} \text{Re } (s - z) \\ \text{Im } (s - z) \end{pmatrix}, \tag{4.14}$$

where

$$A = \begin{pmatrix} \text{Re } q_1 & -\text{Im } q_1 \\ \text{Im } q_2 & \text{Re } q_2 \end{pmatrix}.$$

Now note that $|z| = \|(\text{Re } (z), \text{Im } (z))^T\|_2$ for any $z \in \mathbb{C}$. We use this by solving (4.14) for $(\text{Re } (s - z), \text{Im } (s - z))^T$ and applying the triangle inequality,

$$|s - z| \leq \left\|A^{-1}\right\|_2 |\alpha|.$$

In order to prove (4.12), it remains to show that the determinant of A is $\text{Re } (q_1\overline{q_2}))$. We use the rule for complex numbers that $\text{Re } (a\overline{b}) = \text{Re } a \text{Re } b + \text{Im } a \text{Im } b$, to show that

$$\det \begin{pmatrix} \text{Re } q_1 & -\text{Im } q_1 \\ \text{Im } q_2 & \text{Re } q_2 \end{pmatrix} = \text{Re } q_1 \text{Re } q_2 + \text{Im } q_1 \text{Im } q_2 = \text{Re } (q_1\overline{q_2}).$$

It follows from the definition of singular values that the 2-norm of the inverse of a matrix is the the inverse of the smallest singular value. We have proven (4.13). This completes the proof.

□

This theorem can be used to analyze sensitivity as follows. Consider a nonlinear eigenvalue problem depending on a parameter h, G_h. We wish to apply Theorem 4.6, and let the two nonlinear eigenvalue be let the two nonlinear eigenvalues problems to be compared be $G = G_h$ and $H = G_{h+\Delta}$. Moreover, $s(h) \in \sigma(G_h)$ and $z = s(h+\Delta) \in \sigma(H)$. Note that if $\Delta \to 0$ then in Theorem 4.6, $\zeta_1, \zeta_2 \to s(h)$ and hence $q(\zeta_1) = q(\zeta_2)$. The matrix

$$A = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

for any $a, b \in \mathbb{R}$ has only one singular value $\sigma_{\min}(A) = \sqrt{a^2 + b^2}$. Hence, the singular value in (4.13) is $\sqrt{(\text{Re } q(s(h)))^2 + (\text{Im } q(s(h))^2} = |v^*(I - G'(s(h)))u|$. We note, that in the limit, the formula (4.13) coincides with the upper bound (4.11).

In fact, the sensitivity of individual eigenvalues can be derived with the chain-rule in the following way. We know from [HP05, Corollary 4.2.4] that the spectrum of a parameter dependent matrix is the union of n continuous functions. That is, for a fixed $s \in \mathbb{C}$

$$\sigma(G_h(s)) = \bigcup_{k=1}^{n} g_k(h, s)$$

for some functions g_k continuous in $h \in \mathbb{R}$. Hence, there is a k such that $s = g_k(h, s)$ for the nonlinear eigenvalue problem $s \in \sigma(G(h, s))$. Note that this implicitly defines a function $s(h)$, which derivative we now assume to exist. From the chain-rule,

$$s'(h) = \frac{d}{dh} g_k(h, s(h)) = (g_k)'_h(h, s(h)) + (g_k)'_s(h, s(h))s'(h),$$

where we denote the partial derivatives of g, by $(g)'_h$ and $(g)'_s$ with respect to h and s correspondingly. Rearrangement of terms yields,

$$s'(h) = \frac{(g_k)'_h(h, s(h))}{1 - (g_k)'_s(h, s(h))}.$$

We note that $(g_k)'_s = v^* \frac{dG_h}{ds} u$ and $(g_k)'_s = v^* \frac{dG_h}{dh} u$ with the normalization $v^* u = 1$ from (4.10), where v and u are the left and right eigenvector correspondingly. Hence, the sensitivity of a solution to the nonlinear eigenvalue problem $s \in \sigma(G_h(s))$ is given by the formula

$$s'(h) = \frac{v^* \frac{dG_h}{dh}(s) u}{v^*(I - \frac{dG_h}{ds}(s))u}. \tag{4.15}$$

This is clearly a generalization of the sensitivity formula for the linear case (4.10).

Example 4.7 *We demonstrate the use of the sensitivity formula on the single delay DDE*

$$G(s) = A_0 + A_1 e^{-hs},$$

for the left-most eigenvalues with respect to perturbations in h. The derivatives are

$$\frac{dG}{dh}(s) = -sA_1 e^{-hs} \text{ and } \frac{dG}{ds}(s) = -hA_1 e^{-hs}.$$

Suppose $h > 0$. The sensitivity quotient (4.15) *is*

$$s'_h(h) = \frac{-sv^* A_1 e^{-hs} u}{v^*(I + hA_1 e^{-hs})u} = \frac{-sv^* A_1 u}{v^*(Ie^{hs} + hA_1)u}. \tag{4.16}$$

The real part of the eigenvalues of the DDE have an unbounded negative real part. It is hence interesting to see how the left-most eigenvalues move. Hence, if $\operatorname{Re} s \to -\infty$ *then*

$$s'_h(h) \to -\frac{s}{h}. \tag{4.17}$$

From this we conclude that the larger the negative real part and the smaller the delay is, the larger the sensitivity. Moreover, the left eigenvalues move to the right.

There are some sensitivity results for DDEs with multiple delays. The continuity of the rightmost eigenvalue (spectral abscissa) with respect to changes in the delays was proven in [Dat78] and can be extended to perturbation in the coefficient matrices. See the book by Michiels and Niculescu [MN07b, Theorem 1.14]. The local continuity of each individual eigenvalue can be proven with Rouché's theorem. See Proposition 1.13 in the same book.

4.3.2 Convergence of fixed point iterations

It is natural to consider the fixed point iteration corresponding to the fixed point form as an iterative method to find the solutions of the nonlinear eigenvalue problem. We now wish to prove some theorems on the convergence and convergence order of such methods. The main result (Theorem 4.10) is that the local order of convergence to solution s_* is the largest k for which $G'(s_*)v = \cdots = G^{(k-1)}(s_*)v = 0$, where v is the eigenvector, i.e., $G(s_*)v = s_*v$.

It turns out that the method known as the *method of successive linear problems* (MSLP) in the literature is such a method. The convergence order of MSLP is known to be two for simple eigenvalues. This is local quadratic convergence is a simple consequence of the main theorem.

We will now state and prove a sufficient condition for an iteration $s_{k+1} \in \sigma(G(s_k))$ to have a local convergence domain.

We characterize linear convergence with the following theorem which is a consequence of Lemma 4.3.

Theorem 4.8 *Let $s_{k+1} \in \sigma(G(s_k))$ with* left *eigenvector w_{k+1} for a differentiable parameter dependent matrix G. Suppose $s_* \in \mathbb{C}$ is a fixed point, i.e., $s_* \in \sigma(G(s_*))$ with* right *eigenvector v. Then*

$$|s_{k+1} - s_*| \leq \frac{|w_{k+1}^T G'(\zeta)v|}{|w_{k+1}^T v|}|s_k - s_*|, \tag{4.18}$$

for some ζ on the line-segment between s_{k+1} and s_*, i.e., $\zeta \in l(s_k, s_*)$.

Proof: The proof consists of multiplying with left and right eigenvectors and applying Lemma 4.3. Since $s_{k+1}w_{k+1}^T = w^T G(s_k)$ and $s_*v = G(s_*)v$ the difference is

$$|s_{k+1} - s_*||w_{k+1}^T v| = |w_{k+1}^T(G(s_k) - G(s_*))v|. \tag{4.19}$$

Let φ be the line going through s_* and z, i.e., $\varphi(\theta) = s_k\theta + s_*(1-\theta)$. We define $f(\theta) = w_{k+1}^T G(\varphi(\theta))v$. Implying that $f'(\theta) = w_{k+1}^T G'(\varphi(\theta))v\varphi'(\theta) = w_{k+1}^T G'(\varphi(\theta))v(s_k - s_*)$. Finally, from Lemma 4.3,

$$|w_{k+1}^T(G(s_k) - G(s_*))v| = |f(1) - f(0)| \leq |f'(x)| = |s_k - s_*||w_{k+1}^T G'(\varphi(x))v|, \tag{4.20}$$

for some $x \in (0,1)$. The proof is completed by dividing (4.19) and (4.20) by $|w_{k+1}^T v|$. □

Example 4.9 *We now present a simple example of a convergent fixed point iteration. Consider the set of nonlinear eigenvalue problems*

$$G(s) = \begin{pmatrix} (1+\alpha)s^2 & -\alpha s \\ q\ln(s) & s \end{pmatrix},$$

with the iteration

$$s_{k+1} \in \sigma(G(s_k)). \tag{4.21}$$

For any choice of α and q, $s = 1$ is an eigenvalue with right eigenvector $v = (1,1)^T$ and left eigenvector $u = (0,1)^T$. The derivative is

$$G'(\zeta)v = \begin{pmatrix} 2(1+\alpha)\zeta - \alpha \\ q/\zeta + 1 \end{pmatrix}.$$

At the fixed point, the coefficient in (4.18) is zero when $w^*G'(s)v = 0$. Hence, we should have fast (local) convergence if $q/s + 1 = 0$, i.e., $q = -1$. We consider the specific choice $\alpha = -2$ and $q = -1$, $G'(1)v = (0,0)^T$. Since $u^*v = 1$, there is a neighborhood of $s = 1$ for which $\|G'(\zeta)v\|/|w^*v| < 1$. Thus, there is a convergence region for the iteration $s_{k+1} \in \sigma(G(s_k))$.

s_0	3.0	
$\sigma(G(s_0))$	-8.4229	$2.4229 = s_1$
$\sigma(G(s_1))$	-5.3165	$1.8688 = s_2$
$\sigma(G(s_2))$	-3.0139	$1.3902 = s_3$
$\sigma(G(s_3))$	-1.6292	$1.0868 = s_4$
$\sigma(G(s_4))$	-1.0984	$1.0040 = s_5$
$\sigma(G(s_5))$	-1.0040	$1 + 8.0 \cdot 10^{-6} = s_6$
$\sigma(G(s_6))$	-1.0000	$1 + 3.2 \cdot 10^{-11} = s_7$
$\sigma(G(s_7))$	-1.0000	1

Table 4.1: The iteration (4.21) converges to an accuracy of machine precision in 7 steps for $s_0 = 3.0$.

The speed of convergence is dependent on $G'(\zeta)v$ where v is the eigenvector corresponding to the solution. If $G'(\zeta)v$ is small, then the convergence is expected to be fast. It is somewhat remarkable that it is sufficient for fast convergence that

$G'(s)$ is small (in magnitude) in one direction, i.e., the direction of the eigenvector. This is a property of the set-valued nature of the fixed point iteration. There is no corresponding property for normal (scalar) fixed point iterations.

The quotient $\kappa_e := \frac{1}{|w^* v|}$ (with proper normalization) is known as the *eigenvalue condition number* corresponding to an eigenvalue s of the matrix $G(s)$ with left eigenvector w and right eigenvector v. The convergence is hence expected to be slow if the eigenvalue condition number of $G(s)$ is large unless $|w^* G'(s) v|$ is small.

Theorem 4.8 can be extended to arbitrary order in the following sense.

Theorem 4.10 *Let $s_{k+1} \in \sigma(H(s_k))$ with* left *eigenvector w_{k+1} for a m times differentiable matrix-function $H : \mathbb{C} \to \mathbb{C}^{n\times n}$. Suppose $s_* \in \mathbb{C}$ is a fixed point, i.e., $s_* \in \sigma(H(s_*))$ with* right *eigenvector v for which $0 = H'(s_*)v = H''(s_*)v = \cdots = H^{(m-1)}(s_*)v$. Then*

$$|s_{k+1} - s_*| \leq \left| \frac{w_{k+1}^* H^{(m)}(\zeta) v}{w_{k+1}^* v} \right| |s_k - s_*|^m$$

for some $\zeta \in l(s_k, s_*)$.

Proof: The proof is an by induction over m. For $m = 1$, we have Theorem 4.8. Suppose the theorem holds for $m - 1$. Then

$$|s_{k+1} - s_*| \leq \left| \frac{w_{k+1}^* H^{(m-1)}(\zeta_2) v}{w_{k+1}^* v} \right| |s_k - s_*|^{m-1}$$

for some $\zeta_2 \in l(s_k, s_*)$. Since $H^{(m-1)}(s_*)v = 0$,

$$\begin{aligned} |w_{k+1}^* H^{(m-1)}(\zeta_2) v| &= |w_{k+1}^* H^{(m-1)}(\zeta_2) v - w_{k+1}^* H^{(m-1)}(s_*) v| \\ &= |w_{k+1}^* H^{(m)}(\zeta) v| |\zeta_2 - s_*|, \end{aligned}$$

for some $\zeta \in l(\zeta_2, s_*)$ from Lemma 4.3. From the fact that $|\zeta_2 - s_*| \leq |s_k - s_*|$ and the induction principle, we conclude that the theorem holds. □

We illustrate the theorem by determining the convergence order of Example 4.9.

Example 4.11 (Example 4.9 continued) *We already saw that $u^* G'(s) v = 0$*

if and only if $q = -1$. *The second derivative of* G *is*

$$G''(s)v = \begin{pmatrix} 2(1+\alpha) \\ -q/s^2 \end{pmatrix}.$$

Since $u^*G''(s)v = -q/s^2$, *the local convergence order (in the sense of Theorem 4.10) is* $m = 2$ *if* $q = -1$ *and* $m = 1$ *otherwise.*

We now wish to construct a method with high local convergence order from an (arbitrary) nonlinear eigenvalue problem. Suppose we have a nonlinear eigenvalue problem represented by G. We now constructing a new nonlinear eigenvalue problem (represented by H) using the similarity transformation (Lemma 4.1). We transform the problem with A such that $H'(s)v = 0$, from which we deduce that the convergence is quadratic in the sense of Theorem 4.10.

Suppose G is such that $1 \notin \sigma(G'(s))$ within some region V. It is then possible to do a Cayley similarity transformation using Lemma 4.1 with $A(s) = G'(s)$, $B = I$. We will now show that this choice turns out to give $H'(s_*)v = 0$ as desired. Moreover, the choice corresponds to the Newton's method (on fixed point form) for the scalar case *and* is equivalent to the so-called *method of successive linear problems* (MSLP) [Ruh73] for the non-scalar case (i.e., the nonlinear eigenvalue problem). From Lemma 4.1 we get the transformed problem

$$H(s) = (I - G'(s))^{-1}(G(s) - sG'(s)).$$

The derivative of H is in general

$$H'(s) = (I - A(s))^{-1}\left(G'(s) - A(s) + A'(s)(I - A(s))^{-1}(G(s) - sI)\right), \quad (4.22)$$

and for this particular choice of A, i.e., $A = G'$,

$$H'(s) = (I - G'(s))^{-1}G''(s)(I - G'(s))^{-1}(G(s) - sI).$$

Clearly $H'(s_*)v = 0$ if $s \in \sigma(G)$ and v the corresponding eigenvector. Since, $H''(s_*)v = -(I - G'(s_*))^{-1}G''(s_*)v$ the convergence is in general not more than quadratic. Again, for the scalar case, the iteration $s_{k+1} = H(s_k)$ is the Newton iteration $s_{k+1} = s_k - F(s_k)/F'(s_k)$ with $F(s) = s - H(s)$. Correspondingly for the non-scalar case, we have the iteration $s_{k+1} \in \sigma(H(s_k))$ where s_{k+1} is chosen as the eigenvalue of $H(s_k)$ closest to some given target s_t. As usual, the inverse

in $H(s)$ is not computed explicitly, but instead the corresponding generalized eigenvalue problem is solved in each step.

The local quadratic convergence to simple eigenvalues of MSLP was proven by H. Voss in [Vos04b] using the implicit function theorem. In our context, it is a straightforward consequence of Theorem 4.10 and the fact that $H'(s)v = 0$ and that $w^*v \neq 0$ for simple eigenvalues.

Example 4.12 *It is illustrative to apply the theory above to another method for nonlinear eigenvalue problems. We revisit a method mentioned by Liao, Bai, Lee and Ko [LBLK06] to solve the nonlinear eigenvalue problem $T(s)v = 0$, where*

$$T(s) = K - sM + E(s) \text{ and } E(s) = i\sum_{j=1}^{p}(s - \sigma_j^2)^{\frac{1}{2}}W_j,$$

and all matrices are symmetric and real. The iteration, which is a special case of the so-called self-consistent iteration *(SCI), can be written in our context as the set-valued fixed point iteration*

$$s_{k+1} \in \sigma(M^{-1}(K + E(s_k))).$$

Thus, we immediately see from Theorem 4.8 that the convergence is at most locally linear to eigenvalue s_* unless $w^*M^{-1}E'(s_*)v = 0$. Moreover, the method is is not expected to converge if $w^*M^{-1}E'(s_*)v > 1$ (with normalization $w^*v = 1$).

These results are somewhat pessimistic for SCI. However, they should be interpreted with care, as they only reflect local convergence properties. Global convergence properties are also important in many applications, e.g. sometimes only a very inaccurate initial approximation is available. Moreover, in a complete computation analysis of a method, it is also necessary to investigate how expensive one step of the iteration is.

4.4 Non-local perturbation results and the Bauer-Fike theorem

The local perturbation analysis similar to the previous section can give the behavior of the change of the individual eigenvalues expressed in terms of the eigenvalue and left and right eigenvectors.

However, the local behavior does unfortunately not give all the information relevant in a perturbation analysis. In many cases we do not have the eigenvalue or its eigenvectors available a priori. In particular, the local sensitivity in formula (4.15) depends on the left and right eigenvectors. We wish to do a more qualitative analysis where this information is not necessary. For instance, the following questions are not necessarily answered by the local analysis: Is the spectrum continuous as a set? Since the spectrum is typically unbounded, this is not directly solved by the sensitivity analysis. What is the maximum (supremum) movement of an eigenvalue? Given a bound of the perturbation, in what regions will the eigenvalues move? Can finite perturbation cause eigenvalues to appear somewhere? For which starting values do the corresponding fixed point iterations converge?

We do not aim to answer the questions. Our goal is more modest. We wish to present a way to generalize the very famous perturbation result, known as the Bauer-Fike theorem [BF60]. The generalization is done to nonlinear eigenvalue problems corresponding to (set-valued) contractions.

We present the result by first noting that the nonlinear eigenvalue problem is a set-valued fixed point problem. The real gain of this interpretation is that we can use some of the results in set-valued fixed point theory. We try to adapt known fixed point theory from the field of (mostly) topology (e.g. [KB01]), such that they fit our purposes. For instance, we will see in Theorem 4.18 that parts of the famous *Banach's contraction mapping principle* (Theorem 4.17) generalize nicely to set-valued fixed point problems. More importantly, we make use of a comparison lemma for (contractive) set-valued fixed point problems by Lim [Lim85] and get a generalization of the Bauer-Fike theorem. It is indeed a generalization, as the main theorem reduces to the Bauer-Fike theorem for the constant case. However, it is not applicable to all nonlinear eigenvalue problems, but only those which have a certain contraction property.

Most of the results in this section only apply to a restricted class of problems (when G has a contraction property). Despite this, the analysis is worthwhile, as, to the author's knowledge, this is the first non-local perturbation result with explicit bounds for the set of solutions of nonlinear (non-polynomial) eigenvalue problems (apart from possibly the method to compute the pseudospectra of nonlinear eigenvalue problems in [MGWN06]).

Non-local explicit perturbation bounds for (linear) eigenvalue problems are

typically stated as bounds on some set-valued distances. Two popular distance measures for sets are the *Hausdorff-distance* here denoted d_H (sometimes *spectral variance* in this context) and the *matching distance*, for further discussion on metrics for spectra see the book by Stewart and Sun [SS90]. Here we will focus on the Hausdorff-distance as the Bauer-Fike theorem can be elegantly stated with this metric and the matching distance is difficult to define and interpret for (countable) infinite sets. The Hausdorff-distance d_H is defined as the maximum of the two max-min-distances between two sets (defined in Section 4.4.1). For the linear eigenvalue problem there are several bounds on the change of the spectrum expressed in norms. For instance, the bound in [Els85] (sometimes referred to as Elsner-type bounds) states that the Hausdorff-distance between the spectrum of the two matrices $A, B \in \mathbb{C}^{n\times n}$ is bounded by $d_H(\sigma(A), \sigma(B)) \leq (\|A\| + \|B\|)^{1-1/n}\|A - B\|^{1/n}$. This is an optimal bound in the sense that equality is sometimes attained. However, the bound is very pessimistic for most cases, in particular when the matrices are diagonalizable. In fact, if we assume that the matrices are diagonalizable then the distance can be reduced to a linear condition in $\|A - B\|$, known as the Bauer-Fike theorem $d_H(\sigma(A), \sigma(B)) \leq \max(\kappa_V(A), \kappa_V(B))\,\|A - B\|$ where $\kappa_V(A)$ is the condition number of the eigenvector matrix of A.

Before discussing how the set-valued interpretation of the nonlinear eigenvalue problem can be used, we wish to demonstrate an important property of set-valued mappings not present for (normal) fixed point problems: Successive application of set-valued fixed point problem to a fixed point, does not necessarily yeild a fixed point. For instance, let $s_* \in \mathbb{C}$ be a fixed point of (4.7), i.e., $s_* \in \sigma(G(s_*))$ and suppose $\tilde{s} \in \sigma(G(s_*))$. For normal fixed point problems, $\tilde{s} = s_*$ and as $\tilde{s}$ is a fixed point, for set-valued fixed point problems this is obviously not necessarily the case. This phenomenon was characterized (mostly for set-valued contractions) by Nadler in [Nad69].

Example 4.13 *We show an example of how successive application of set-valued fixed point problems differs from the single-valued case as mentioned in [Nad69]. Consider the nonlinear eigenvalue problem*

$$G(s) = \begin{pmatrix} s^3 - 6 & 0 \\ 0 & 4s - 9 \end{pmatrix}.$$

The solutions, i.e., the fixed points of $s \in \sigma(G(s))$ are $\sigma(G) = \{2, 3, -1 \pm \sqrt{2}i\}$.

For scalar fixed point problem, one would expect that if $s \in \sigma(G)$ and $\tilde{s} \in \sigma(G(s))$ $\Rightarrow \tilde{s} \in \sigma(G)$. This is clearly not the case. We now define $\sigma(G(\cdot))$ for set arguments in the natural way $\sigma(G(S)) = \cup_{s\in S}\sigma(G(s))$. One would expect that $\sigma(G)$ is related to $\sigma(G(\sigma(G)))$ or $\sigma(G(\sigma(G(\sigma(G(\cdots(\sigma(G))\cdots)))))))$. Here $\sigma(G(\sigma(G))) = \sigma(G)\cup\{-1, 21, -13\pm i\sqrt{32}i\}$. Except for the containment, there seems to be very little relation between the fixed points $\sigma(G)$ and the additional values.

4.4.1 The Bauer-Fike theorem

In the literature, the following form of the Bauer-Fike theorem[1] is often used. Essentially, it bounds the change of eigenvalues in terms of norm of the difference and condition number of the eigenvector matrix.

Theorem 4.14 ([BF60]) *If $\lambda_1 \in \sigma(A_1)$ and $\lambda_2 \in \sigma(A_2)$ then*

$$\begin{aligned} a) \qquad & \operatorname{dist}(\lambda_1, \sigma(A_2)) \le \kappa(V_2)\|A_1 - A_2\| \\ b) \qquad & \operatorname{dist}(\lambda_2, \sigma(A_1)) \le \kappa(V_1)\|A_1 - A_2\| \end{aligned}$$

where $\operatorname{dist}(\lambda, S) = \min_{s\in S}|\lambda - s|$ assuming A_1 and A_2 are diagonalizable with $A_i = V_i^{-1}\Lambda_i V_i$ and $\kappa_V(V) = \|V\|\|V^{-1}\|$.

For brevity, we will denote the condition number of the eigenvector matrix corresponding to the diagonalizable matrix A by $\kappa_V(A)$. As usual, we say that $\kappa_V(A) = \infty$ when A is not diagonalizable.

Geometrically, the theorem states that the union of discs with radius $\kappa(A_2)\|A_1 - A_2\|$ centered at $\sigma(A_2)$ covers the spectrum of A_1. Note that Bauer-Fike theorem yields different perturbation information depending on the direction it is applied, i.e., it also holds that the union of discs with radius $\kappa_V(A_1)\|A_1 - A_2\|$ centered at $\sigma(A_1)$ covers the spectrum of A_2. This is consistent with the fact that the max-min distance[2] from one set to the other is not symmetric and therefore not a metric. It is hence natural to consider the maximum of the two max-min distances, which is a metric. This is in fact *the Hausdorff distance* or *Hausdorff*

[1] In fact, the theorem is somewhat unfairly called the Bauer-Fike theorem. The results of Bauer and Fike [BF60] are much more general than the partial result presented here and commonly described in the literature (as pointed out in [SS90, p 177]).

[2] The max-min distance for eigenvalue perturbation theory is often called *spectral variance*.

metric (following the notation in [HP05, Chapter IV]), defined by

$$d_H(S_1, S_2) := \max\left(\max_{s_1 \in S_1} \operatorname{dist}(s_1, S_2), \max_{s_2 \in S_2} \operatorname{dist}(s_2, S_1)\right),$$

for finite sets $S_1,\ S_2$, where as usual $\operatorname{dist}(\cdot,\cdot)$ is the minimum distance, $\operatorname{dist}(s, S_1) := \min_{s_1 \in S_1} |s - s_1|$. With a slight loss of generality the Bauer-Fike theorem can be formulated elegantly using the Hausdorff metric.

Theorem 4.15 ([BF60] see also [SS90, Theorem IV.3.3]) *Suppose $A_1, A_2 \in \mathbb{C}^{n\times n}$ are diagonalizable. Then,*

$$d_H(\sigma(A_1), \sigma(A_2)) \leq \max(\kappa_V(A_1), \kappa_V(A_2))\|A_1 - A_2\|.$$

4.4.2 Contraction mappings in set-valued fixed point theory

We pointed out earlier that the nonlinear eigenvalue problem in fixed point form is a so-called *set-valued fixed point problem.* We will now state some known theorems for set-valued fixed point problems which will be used in the next subsection to generalize the Bauer-Fike theorem.

We first state some definitions and fundamental principles for normal fixed point problems $\varphi(x) = x$. Contractivity and Banach's contraction mapping principle is often used to prove convergence and to analyze the convergence order of fixed point iterations. Since this is also our goal, we first define contraction in a connected region $V \subset \mathbb{C}$ and then state Banach's contraction mapping principle. We follow the terminology in [KB01] and refer the reader to this book for proofs of the theorems.

Definition 4.16 *A mapping $\varphi : V \to V$ is said to be* lipschitzian *(in a region V) if there is a contraction constant $L \geq 0$ such that $|\varphi(x) - \varphi(y)| \leq L|x - y|$ for all $x, y \in V$. The smallest such value L is called the* Lipschitz constant *and the mapping is said to be a* contraction mapping *if this value is less than one.*

In the following φ^n denotes the successive application of φ, i.e., $\varphi^n(s) = \varphi(\varphi^{n-1}(s))$.

Theorem 4.17 (Banach's Contraction Mapping Principle) *Let φ be a contraction mapping with contraction constant $L < 1$. Then the mapping φ has a unique fixed point s_*. Moreover, for each $s \in V$,*

$$\lim_{n\to\infty} \varphi^n(s) = s_*, \tag{4.23}$$

and for each $s \in M$,

$$|\varphi^n(s) - s_0| \leq \frac{L^n}{1-L}|s - \varphi(s)|,\ n \in \mathbb{Z}_+. \tag{4.24}$$

Parts of the Banach's contraction principle generalizes to set-valued fixed point problems here denoted by,

$$s \in T(s),$$

where T is a set-valued mapping. In the later sections on perturbation theory we will apply the results to $T(s) = \sigma(G(s))$ or $T(s) = \sigma(G(s)) \cap V$. We saw in Example 4.13 that successive application of $T(s_*)$ for set-valued fixed points introduced additional points, which is one of the reasons why it is not obvious how the generalization of the contraction mapping principle should be done.

In the literature the set-valued generalization of Theorem 4.17 is only an existence result. The following theorem does not guarantee uniqueness and does not have corresponding formulas (4.23) and (4.24).

Theorem 4.18 ([Nad69]) *Let M be the collection of all nonempty bounded closed subsets of V endowed with the Hausdorff metric. Suppose $T : V \to M$ is a contraction in the sense that for some $L < 1$:*

$$d_H(T(x)), T(y)) \leq L|x - y|, x, y \in V,$$

Then there exists a point $s_* \in V$ such that $s_* \in T(s_*)$.

We will also use a comparison lemma by Lim [Lim85]. The result gives a bound of the (Hausdorff) distance of the set of fixed points for two set-valued contraction mappings. We denote the set of fixed points of the set-valued fixed point problem $x \in T(x)$ with $F(T)$.

Theorem 4.19 ([Lim85]) *Suppose M is the collection of all nonempty bounded subsets of V endowed with the Hausdorff metric, and let $T_i : M \to M$, $i = 1, 2$, be two contraction mappings each having Lipschitz constant $L < 1$. Then,*

$$d_H(F(T_1), F(T_2)) \leq \frac{1}{1-L} \sup_{s \in V} d_H(T_1(s), T_2(s)).$$

4.4.3 A Bauer-Fike theorem for nonlinear eigenvalue problems

The idea is now to combine Lim's comparison lemma (Theorem 4.19) and the Bauer-Fike theorem (Theorem 4.15) in order to get non-local perturbation results for nonlinear eigenvalue problems. We will first do so by applying Lim's comparison lemma to the set-valued mappings $T_i := \sigma(G_i(s)) \cap V = \sigma_V(G_i(s))$. We use the notation $\sigma_V(G) := \{s \in V : s \in \sigma(G(s)) \cap V\}$.

Under the assumption that the set-valued fixed point problems T_1 and T_2 are contractions we can apply Lim's comparison lemma yielding the following comparison theorem. We will see how it reduces to the Bauer-Fike theorem in Corolloary 4.23.

Theorem 4.20 *Let $V \subset \mathbb{C}$ be a given convex subset of $\mathbb{C}$. Suppose that $G_i : \mathbb{C} \to \mathbb{C}^{n \times n}$, $i = 1, 2$ are differentiable in V and that there are constants and $\varepsilon \geq 0$ and $\kappa > 0$ such that, $\kappa\varepsilon < 1$, $\varepsilon \geq \sup_{s \in V} \|G_i'(s)\|$, and*

$$d_H(\sigma_V(G_1(s)), \sigma_V(G_2(s))) \leq \kappa \|G_1(s) - G_2(s)\| \tag{4.25}$$

for all $s \in V$ and for any $s_1, s_2 \in V$,

$$d_H(\sigma_V(G_i(s_1)), \sigma_V(G_i(s_2))) \leq \kappa \|G_i(s_1) - G_i(s_2)\|. \tag{4.26}$$

Then,

$$d_H(\sigma_V(G_1), \sigma_V(G_2)) \leq \frac{\kappa}{1-\kappa\varepsilon} \sup_{s \in V} \|G_1(s) - G_2(s)\|. \tag{4.27}$$

Proof: We wish to apply Theorem 4.19 and therefore first want to bound the contraction constant. It follows from the mean value theorem that $\|G_i(x) - G_i(y)\| \leq \|G_i'(\zeta)\||x - y|$ for some $\zeta \in l(x, y)$ which is also in V because V is convex. Hence, from (4.26) we have that for any $s_1, s_2 \in V$,

$$d_H(\sigma_V(G_i(s_1)), \sigma_V(G_i(s_2))) \leq \kappa\varepsilon |s_1 - s_2|$$

and the set-valued fixed point problems $T_i(s) = \sigma_V(G_i(s))$ are contractions because $L = \kappa\varepsilon < 1$ by assumption. We can now apply Theorem 4.19 to T_i, since $T_i(s) = \sigma_V(G_i(s))$ is bounded (even if V is unbounded). It yields that

$$d_H(\sigma_V(G_1), \sigma_V(G_2)) \leq \frac{1}{1-\kappa\varepsilon} \sup_{s\in V} d_H(\sigma_V(G_1(s)), \sigma_V(G_2(s))).$$

We get (4.27) and complete the proof by applying assumption (4.25). □

It is tempting to directly use the Bauer-Fike theorem to bound κ and get (4.25) and (4.26). This is however not in general possible because (4.25) and (4.26) involve the restriction of the spectrum, i.e., $\sigma_V(G_i(s))$ and not $\sigma(G_i(s))$. The distance of the restriction of two sets S_1, S_2 is sometimes larger and sometimes smaller than the distance between S_1 and S_2, i.e., $d_H(S_1, S_2) \not\leq d_H(S_1 \cap V, S_2 \cap V)$ nor $d_H(S_1, S_2) \not\geq d_H(S_1 \cap V, S_2 \cap V)$ in general. This prevents us from directly applying the Bauer-Fike theorem. However, clearly if $\sigma_V(G_i(s)) = \sigma(G_i(s))$ then the Bauer-Fike theorem can be applied. We show this with two examples, and summarize the result as a corollary. With the examples we aim to illustrate how to apply Theorem 4.20. There are certainly other more efficient and less pessimistic ways to do a perturbation analysis of the problems.

Example 4.21 *Consider the small nonlinear perturbation of a matrix,*

$$G_1(s) = A_1,$$

$$G_2(s) = A_1 + A_2\cos(|s|).$$

Note that $G_2(s)$ is differentiable in $s \in \mathbb{C}$. Clearly, if we set $\varepsilon = \|A_2\|$, $\varepsilon \geq \sup_{s\in\mathbb{C}} \|G_i'(s)\|$. We can numerically verify that $\kappa_V(G_2(s)) < 1.08$ if

$$A_1 = \begin{pmatrix} 7 & 10 \\ 8 & -15 \end{pmatrix}, \; A_2 = \frac{1}{10}\begin{pmatrix} -1 & -1 \\ 0.7 & 0.6 \end{pmatrix}.$$

Equations (4.25) *and* (4.26) *are fulfilled for $\kappa = 1.08$ from the Bauer-Fike Theorem 4.15. Hence, Theorem 4.20 yields that*

$$d_H(\sigma(G_1), \sigma(G_2)) \leq \frac{\kappa\|A_2\|}{1-\kappa\|A_2\|} < 0.223.$$

Since Theorem 4.20 is a non-local perturbation result, we have now proven that all elements of $\sigma(G_2)$ are indeed small perturbations of $\sigma(A_1)$, i.e., all eigenvalues corresponding to G_2 lie within the discs of radius 0.223 *centered around the*

eigenvalues of A_1, $\sigma(A_1) \approx \{10.177, -18.177\}$. *It is not generally the case that* all *solutions of a perturbed problem are close to the unperturbed problem even if the nonlinear perturbation is small in magnitude. This phenomenon (which we have now managed to exclude for this example) is typically explained, loosely speaking, as a perturbation of the eigenvalues from infinity.*

Example 4.22 *Now suppose* $V = \mathbb{R}$ *and* $G_1(s)$ *and* $G_2(s)$ *normal matrices for any* $s \in \mathbb{R}$. *Since* (4.25) *and* (4.26) *hold for* $\kappa = 1$, *Theorem 4.20 can be applied if* $\|G_1'(s)\|$ *and* $\|G_2'(s)\|$ *are less than one for every* $s \in \mathbb{R}$. *Consider the two nonlinear eigenvalue problems*

$$G_1(s) = A_0 + A_1 \sin(s)$$

$$G_2(s) = A_0 + A_1 \sin(\alpha s)$$

where $\alpha > 1$. *Then,*

$$\sup_{s\in\mathbb{R}} \|G_1(s) - G_2(s)\| = \sup_{s\in\mathbb{R}} \|A_1\| |\sin(s) - \sin(\alpha s)| \leq 2\|A_1\|.$$

If we let $\varepsilon = \alpha\|A_1\|$ *and assume that* $\alpha\|A_1\| < 1$ *then*

$$d_H(\sigma_{\mathbb{R}}(G_1), \sigma_{\mathbb{R}}(G_2)) \leq \frac{2\|A_1\|}{1 - \alpha\|A_1\|}.$$

independent of A_0.

The two examples above are both cases where $\sigma_V(G(s)) = \sigma(G(s))$ $s \in V$. This is summarized in the following corollary.

Corollary 4.23 *Let* $V \subset \mathbb{C}$ *be a given convex subset of* $\mathbb{C}$. *Suppose that* $G_i : \mathbb{C} \to \mathbb{C}^{n\times n}$, $i = 1, 2$ *are differentiable in* V, $\sigma(G_i(s)) \subset V$ *for all* $s \in V$, *and that there are* $\varepsilon \geq 0$ *and* $\kappa > 0$ *such that* $\kappa\varepsilon < 1$, $\varepsilon \geq \sup_{s\in V} \|G_i'(s)\|$ *and* $\kappa \geq \sup_{s\in V} \kappa_V(G_i(s))$, *for* $i = 1, 2$. *Then,*

$$d_H(\sigma_V(G_1), \sigma_V(G_2)) \leq \frac{\kappa}{1 - \kappa\varepsilon} \sup_{s\in V} \|G_1(s) - G_2(s)\|. \tag{4.28}$$

Proof: By assumption $\sigma(G_i(s)) \subset V$ for all $s \in V$ and $\sigma_V(G_i(s))) = \sigma(G_i(s))$. The left hand sides in (4.25) and (4.26) can be estimated by the Bauer-Fike theorem, Theorem 4.15 yielding the bound if $\kappa = \max_{i=1,2} \sup_{s\in V} \kappa_V(G_i(s))$. □

Appendix A

Appendix

A.1 Linearization of polynomial eigenproblems

The problem of determining $\lambda \in \mathbb{C}$ and $v \in \mathbb{C}\backslash\{0\}$ such that

$$P(\lambda)v = \left(A_0 + A_1\lambda + \cdots + A_N\lambda^N\right) v = 0, \tag{A.1}$$

where $A_0, \ldots, A_N \in \mathbb{C}^{n\times n}$ is often called a polynomial eigenvalue problem (PEP). The PEP (A.1) includes many important sub-problems. If $N = 1$ (A.1) is a standard generalized eigenvalue problem (GEP), if $N = 2$ the (A.1) is called a quadratic eigenvalue problem and if $n = 1$ then (A.1) is a polynomial root-finding problem. The following known results follow the style of [MMMM06a] and [MMMM06b].

The classical way to analyze or solve a PEP (A.1) is by transforming it to a standard generalized eigenvalue problem. The most common approach is to consider the matrix pencil $L(\lambda) = \lambda X + Y$ such that there are $E(\lambda)$, $F(\lambda)$ and

$$E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} P(\lambda) & 0 \\ 0 & I \end{pmatrix}.$$

Among this class of problems the first and second companion linearizations are the most common. The first companion linearization $C_1(\lambda) = \lambda X_1 + Y_1$ is given

by

$$X_1 = \begin{pmatrix} A_N & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix}, Y_1 = \begin{pmatrix} A_{N-1} & A_{N-2} & \cdots & A_0 \\ -I & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ & & -I & \vdots \end{pmatrix} \tag{A.2}$$

and the second companion form $C_2(\lambda) = \lambda X_2 + Y_2$ by

$$X_2 = X_1,\ Y_1 = \begin{pmatrix} A_{N-1} & -I & & 0 \\ A_{N-2} & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & -I \\ A_0 & 0 & \cdots & 0 \end{pmatrix}.$$

The two companion linearizations are by no means the only way to transform the PEP to a GEP of larger dimension. Mackey, et al. [MMMM06b] construct vector-spaces of (potential) linearizations L_1 and L_2 generalizing the first and second companion form. This turns out to be advantageous for many applications, since a correct choice of linearizations L_1 and or L_2 can preserve properties of the original problem. This includes different types of palindromic and hermitian structures [MMMM06a]. As mentioned in Section 3.3.1, a linearization which preserves the eigenvalue pairing of the polynomial eigenvalue problem in Theorem 3.29 is given in [FMMS07].

Linearization can be used to numerically solve the PEP, by applying, e.g., a general purpose eigenvalue solver such as the QZ-algorithm implemented in `eig` in MATLAB (which is based on LAPACK library). There are also approaches which are not based on linearization, such as the second order Arnoldi [BS05] and generalizations of Jacobi-Davidson [SBFvdV96]. Moreover, any of the methods for nonlinear eigenvalue problems discussed in Section 2.3 are applicable to polynomial eigenvalue problems.

In this work we almost exclusively use the first companion form (A.2) to numerically solve the polynomial eigenvalue problem.

Bibliography

[ACG+99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, and A. Marchetti-Spaccamela, *Complexity and approximation*, Springer, 1999. (Cited on page 143.)

[ACS05] A. Abate, M. Chen, and S. Sastry, *New congestion control schemes over wireless networks: Delay sensitivity analysis and simulations*, Proc. 16th IFAC World Congress, Prague, CZ, Jul. 2005, 2005. (Cited on page 17.)

[AH80] C. E. Avellar and J. Hale, *On the zeros of exponential polynomials*, J. Math. Anal. Appl. 73 (1980), no. 2, 434–452. (Cited on page 113.)

[AP05] A. Ahlborn and U. Parlitz, *Multiple delay feedback control a delay method for stabilizing fixed points*, ENOC-2005, Eindhoven, 2005, 10-263. (Cited on page 79.)

[Atk72] F. Atkinson, *Multiparameter Eigenvalue Problems. volume I: Matrices and compact operators*, Academic press, New York, 1972. (Cited on pages 131 and 132.)

[AU03] F. M. Asl and A. G. Ulsoy, *Analysis of a system of linear delay differential equations*, J. Dyn. Sys. and Contr. 125 (2003), 215–223. (Cited on page 17.)

[Bam07] M. Bambi, *Endogenous growth and time-to-build: The AK case*, J. Economic Dyn. and Contr. (2007), doi:10.1016/j.jedc.2007.04.002, in press. (Cited on page 17.)

[Bau85] H. Baumgärtel, *Analytic perturbation theory for matrices and operators*, Operator Theory: Advances and applications Vol 15., Birkhäuser Verlag, 1985. (Cited on page 147.)

[BB90] J. Beachy and W. Blair, *Abstract algebra*, 2 ed., Waveland Press, 1990. (Cited on page 142.)

[BC63] R. Bellman and K. L. Cooke, *Differential-Difference Equations*, Academic Press, 1963. (Cited on pages 5, 83, 84, and 85.)

[BC94] J. Bélair and S. A. Campbell, *Stability and bifurcations of equilibria in a multiple-delayed differential equation*, SIAM J. Appl. Math. 54 (1994), no. 5, 1402–1424. (Cited on pages 79 and 85.)

[Bet07a] M. Betcke, *Iterative projection methods for symmetric nonlinear eigenvalue problems with applications*, Ph.D. thesis, Technical University Hamburg-Harburg, 2007. (Cited on pages 51 and 55.)

[Bet07b] T. Betcke, *Optimal scaling of generalized and polynomial eigenvalue problems*, Tech. report, Manchester Institute for Mathematical Sciences, 2007. (Cited on page 150.)

[BF60] F. Bauer and C. Fike, *Norms and exclusion theorems*, Numer. Math. 2 (1960), 137–141. (Cited on pages 163, 165, and 166.)

[Bic48] W. G. Bickley, *Difference and associated operators, with some applications*, J. Math. Phys. 27 (1948), 183–192, issn:0097-1421. (Cited on page 35.)

[BK02] E. Beretta and Y. Kuang, *Geometric stability switch criteria in delay differential systems with delay dependent parameters*, SIAM J. Math. Anal. 33 (2002), no. 5, 1144–1165. (Cited on pages 74, 85, and 86.)

[BL02] E.-K. Boukas and Z.-K. Liu, *Deterministic and stochastic time delay systems*, Birkhäuser, 2002. (Cited on page 5.)

[BM00] A. Bellen and S. Maset, *Numerical solution of constant coefficient linear delay differential equations as abstract Cauchy problems*, Numer. Math. 84 (2000), no. 3, 351–374. (Cited on pages 15 and 42.)

[BMB+04] E. A. Butcher, H. Ma, E. Bueler, V. Averina, and Z. Szabo, *Stability of linear time-periodic delay-differential equations via chebyshev polynomials*, Int. J. Numer. Methods Eng. 59 (2004), no. 7, 895–922. (Cited on page 37.)

[BMV04] D. Breda, S. Maset, and R. Vermiglio, *Computing the characteristic roots for delay differential equations*, IMA J. Num. An. 24 (2004), 1–19. (Cited on pages 42 and 43.)

[BMV05a] ———, *Efficient computation of stability charts for linear time delay systems*, Proc. ASME-IDETC/CIE, 2005. (Cited on page 76.)

[BMV05b] ———, *Pseudospectral differencing methods for characteristic roots of delay differential equations*, SIAM J. Sci. Comput. 27 (2005), no. 2, 482–495. (Cited on pages 42, 43, and 44.)

[BMV06a] ———, *Numerical computation of characteristic multipliers for linear time periodic coefficients delay differential equations*, IFAC Workshop on Time Delay Systems (TDS'06) L'Aquila, Italy, 2006. (Cited on page 37.)

[BMV06b] ———, *Pseudospectral approximation of eigenvalues of derivative operators with non-local boundary conditions*, Applied Numerical Mathematics 56 (2006), 318–331. (Cited on page 42.)

[Boe95] F. Boese, *Stability criteria for second-order dynamical systems involving several time delays*, SIAM J. Math. Anal. 26 (1995), no. 5, 1306–1330. (Cited on page 85.)

[Boe98] ———, *Stability with respect to the delay: On a paper of K. L. Cooke and P. van den Driessche*, J. Math. Anal. Appl. 228 (1998), no. 2, 293–321. (Cited on pages 85 and 86.)

[BP07] C. Bonnet and J. Partington, *Stabilization of some fractional delay systems of neutral type*, Automatica 43 (2007), 2047–2053. (Cited on pages 45, 85, and 88.)

[Bre04] D. Breda, *Numerical computation of characteristic roots for delay differential equations*, Ph.D. thesis, Università degli Studi di Padova, 2004. (Cited on pages 37 and 42.)

[Bre06] ———, *Solution operator approximations for characteristic roots of delay differential equations*, Appl. Numer. Math. 56 (2006), 305–317. (Cited on pages 35 and 36.)

[BS05] Z. Bai and Y. Su, *SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl. 26 (2005), no. 3, 640–659. (Cited on pages 47, 100, and 172.)

[BSS98] L. Blum, F. Shub, and S. Smale, *Complexity and real computation*, New York: Springer, 1998. (Cited on pages 141 and 143.)

[BT00] V. D. Blondel and J. N. Tsitsiklis, *A survey of computational complexity results in systems and control*, Automatica 36 (2000), no. 9, 1249–1274. (Cited on page 143.)

[Bue04] E. Bueler, *Chebyshev collocation for linear, periodic ordinary and delay differential equations: A posteriori estimates*, Tech. report, Dept. of Math. Stat., Univ. Alaska, Fairbanks, 2004, arXiv:math.NA/0409464. (Cited on page 37.)

[Bue07] ———, *Error bounds for approximate eigenvalues of periodic-coefficient linear delay differential equations*, SIAM J. Num. Analysis 45 (2007), no. 6, 2510–2536. (Cited on pages 37, 38, and 44.)

[BV04] T. Betcke and H. Voss, *A Jacobi-Davidson type projection method for nonlinear eigenvalue problems*, Future Generation Computer Systems 20 (2004), no. 3, 363–372. (Cited on page 54.)

[BV07] M. Betcke and H. Voss, *Stationary Schrödinger equations governing electronic states of quantum dots in the presence of spinorbit splitting*, Appl. Math. 52 (2007), 267 – 284. (Cited on page 51.)

[BZ03] A. Bellen and M. Zennaro, *Numerical methods for delay differential equations*, Oxford university press, 2003. (Cited on pages 5 and 42.)

[CCH+96] R. Corless, G. Connet, D. Hare, D. J. Jeffrey, and D. Knuth, *On the Lambert W function*, Adv. Comput. Math. 5 (1996), 329–359. (Cited on pages 16, 17, and 20.)

[CG82] K. L. Cooke and Z. Grossman, *Discrete delay, distributed delay and stability switches*, J. Math. Anal. Appl. 86 (1982), 592–627. (Cited on pages 85 and 108.)

[CGN95] J. Chen, G. Gu, and C. N. Nett, *A new method for computing delay margins for stability of linear delay systems*, Syst. Control Lett. 26 (1995), no. 2, 107–117. (Cited on pages 85, 89, 105, 107, 109, 112, 131, 136, 137, and 139.)

[Chi88] J. Chiasson, *A method for computing the interval of delay values for which a differential-delay system is stable*, IEEE Trans. Autom. Control 33 (1988), no. 12, 1176–1178. (Cited on page 132.)

[Chu03] E. K.-W. Chu, *Perturbation of eigenvalues for matrix polynomials via the Bauer-Fike theorems*, SIAM J. Matrix Anal. Appl. 25 (2003), no. 2, 551–573. (Cited on page 149.)

[CM02a] Y. Chen and K. L. Moore, *Analytical stability bound for a class of delayed fractional-order dynamic systems*, Nonlinear Dyn. 29 (2002), 191–200. (Cited on page 17.)

[CM02b] ———, *Analytical stability bound for delayed second-order systems with repeating poles using Lambert function W*, Automatica 38 (2002), no. 5, 891–895. (Cited on page 17.)

[CNW06] S. A. Campbell, I. Ncube, and J. Wu, *Multistability and stable asynchronous periodic oscillations in a multiple-delayed neural system*, Physica D 214 (2006), 101–119. (Cited on page 79.)

[CR01] J. Cullum and A. Ruehli, *Pseudospectra analysis, nonlinear eigenvalue problems, and studying linear systems with delays*, BIT 41 (2001), no. 2, 265–281. (Cited on page 147.)

[CS00] B. Cahlon and D. Schmidt, *On stability of systems of delay differential equations*, J. Comput. Appl. Math. 117 (2000), no. 2, 137–158. (Cited on page 85.)

[CS04] ———, *Stability criteria for certain second-order delay differential equations with mixed coefficients*, J. Comput. Appl. Math. 170 (2004), no. 1, 79–102. (Cited on page 85.)

[CvdD86] K. L. Cooke and P. van den Driessche, *On zeroes of some transcendental equations*, Funkc. Ekvacioj, Ser. Int. 29 (1986), 77–90. (Cited on pages 85 and 86.)

[Dat78] R. Datko, *A procedure for determination of the exponential stability of certain differential-difference equations*, Q. Appl. Math. 36 (1978), 279–292. (Cited on pages 73 and 157.)

[DB08] G. Dahlquist and Å. Björck, *Numerical methods in scientific computing volume I*, SIAM, Philadelphia, 2008. (Cited on page 35.)

[DvGVW95] O. Diekmann, S. A. van Gils, S. M. Verduyn Lunel, and H.-O. Walther, *Delay equations. functional-, complex-, and nonlinear analysis*, Applied Mathematical Sciences. 110. New York, NY: Springer-Verlag, 1995. (Cited on pages 5, 6, 25, 27, 39, and 73.)

[EA06] C. Ebenbauer and F. Allgöwer, *Stability analysis for time-delay systems using Rekasius's Substitution and sum of squares*, Proceedings of the 45th Conference on Decision & Control, San Diego, 2006, pp. 5376–5381. (Cited on page 88.)

[EJ92] J.-C. Evard and F. Jafari, *A complex rolle's theorem*, Am. Math. Mon. 99 (1992), no. 9, 858–861. (Cited on pages 154 and 155.)

[ELR02] K. Engelborghs, T. Luzyanina, and D. Roose, *Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL*, ACM Transactions on Mathematical Software 28 (2002), no. 1, 1–24. (Cited on page 15.)

[Els85] L. Elsner, *An optimal bound for the spectral variation of two matrices*, Linear Algebra Appl. 71 (1985), 77–80. (Cited on page 164.)

[Eng00] K. Engelborghs, *DDE-BIFTOOL: a Matlab package for bifurcation analysis of delay differential equations*, Tech. report, Department of Computer Science, K.U. Leuven, 2000. (Cited on pages 25, 29, and 52.)

[EO06] A. F. Ergenc and N. Olgac, *Extended Kronecker summation for determining the kernel and offspring of LTI systems with multiple delays*, Proc. of IFAC TDS, L'Aquila, 2006. (Cited on page 89.)

[EOF07] A. F. Ergenc, N. Olgac, and H. Fazelina, *Extended Kronecker summation for cluster treatment of LTI systems with multiple delays*, SIAM J. Control Optimization 46 (2007), no. 1, 143–155. (Cited on pages 85, 89, 90, and 131.)

[FK06] H. Fassbender and D. Kressner, *Structured eigenvalue problems*, GAMM Mitteilungen 29 (2006), no. 2, 297 – 318, Themenheft Applied and Numerical Linear Algebra, Part II. (Cited on page 149.)

[FMMS07] H. Fassbender, N. Mackey, S. Mackey, and C. Schröder, *A structured polynomial eigenproblem arising in the analysis of time delay systems and related polynomial eigenproblems*, Tech. report, TU Braunschweig, 2007, submitted to ETNA. (Cited on pages 99, 128, and 172.)

[FNC05] P. Fu, S.-I. Niculescu, and J. Chen, *Stability of linear neutral time-delay systems: Exact conditions via matrix pencil solutions*, Proc. Am. Contr. Conf., Portland, 2005. (Cited on page 131.)

[FNC06] ———, *Stability of linear neutral time-delay systems: Exact conditions via matrix pencil solutions*, IEEE Trans. Autom. Control 51 (2006), no. 6, 1063–1069. (Cited on pages 85, 89, 120, 121, 122, 123, 125, 131, 137, 139, and 140.)

[FR86] H. Freedman and V. Sree Hari Rao, *Stability criteria for a system involving two time delays*, SIAM J. Appl. Math. 46 (1986), no. 4, 552–560. (Cited on pages 84 and 85.)

[Fri06] E. Fridman, *Stability of systems with uncertain delays: A new completeLyapunov-Krasovskii functional*, IEEE Trans. Autom. Control 51 (2006), no. 5, 885–890. (Cited on page 83.)

[FSO07] H. Fazelina, R. Sipahi, and N. Olgac, *Stability robustness analysis of multiple time-delayed systems using "building block" concept*, IEEE Trans. Autom. Control 52 (2007), no. 5, 799–810. (Cited on page 88.)

[GCS08] D. Ghosh, A. R. Chowdhury, and P. Saha, *Multiple delay rossler system–bifurcation and chaos control*, Chaos, Solitons & Fractals 35 (2008), 472–485. (Cited on page 17.)

[GKC03] K. Gu, V. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*, Control Engineering. Boston, MA: Birkhäuser, 2003. (Cited on pages 5, 76, 78, 81, 83, 132, 141, and 143.)

[GMO+06] M. Gu, E. Mengi, M. Overton, J. Xia, and J. Zhu, *Fast methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl. 28 (2006), no. 2, 477–502. (Cited on page 128.)

[GN00] K. Gu and S.-I. Niculescu, *Additional dynamics in transformed time-delay systems*, IEEE Trans. Autom. Control 45 (2000), no. 3, 572–575. (Cited on page 74.)

[GNC05] K. Gu, S.-I. Niculescu, and J. Chen, *On stability crossing curves for general systems with two delays*, J. Math. Anal. Appl. 311 (2005), no. 1, 231–253. (Cited on pages 74, 82, 85, 86, 93, and 94.)

[GvdV00] G. Golub and H. A. van der Vorst, *Eigenvalue computation in the 20th century*, J. Comput. Appl. Math. 123 (2000), 35–65. (Cited on page 145.)

[GW06] K. Green and T. Wagenknecht, *Pseudospectra and delay differential equations*, J. Comput. Appl. Math. 196 (2006), no. 2, 567–578. (Cited on page 148.)

[Had67] K. Hadeler, *Mehrparametrige und nichtlineare Eigenwertaufgaben*, Arch. Ration. Mech. Anal. 27 (1967), 306–328. (Cited on page 50.)

[Hal91] J. Hale, *Dynamics and delays*, Delay differential equations and dynamical systems, Proc. Conf., Claremont/CA (USA) 1990, Lect. Notes Math. 1475, 16-30 (1991)., 1991. (Cited on pages 85 and 86.)

[Hay50] N. Hayes, *Roots of the transcendental equation associated with a certain difference-differential equation*, J. Lond. Math. Soc. 25 (1950), 226–232. (Cited on page 84.)

[Hay05] B. Hayes, *Why W?*, American Scientist 93 (2005), no. 2, 104–108. (Cited on page 16.)

[HC05] C. Hwang and Y.-C. Cheng, *A note on the use of the Lambert W function in the stability analysis of time-delay systems*, Automatica 41 (2005), no. 10, 1979–1985. (Cited on page 17.)

[HC06] J. Heffernan and R. Corless, *Solving some delay differential equations with computer algebra*, Math. Scientist 31 (2006), no. 1, 21–34. (Cited on page 17.)

[HH93] J. Hale and W. Huang, *Global geometry of the stable regions for two delay differential equations*, J. Math. Anal. Appl. 178 (1993), no. 2, 344–362. (Cited on pages 74, 78, 85, and 86.)

[Hig01] N. J. Higham, *Evaluating Padé approximants of the matrix logarithm*, SIAM J. Matrix Anal. Appl. 22 (2001), no. 4, 1126–1135. (Cited on page 35.)

[Hig06] ———, *Functions of matrices*, Handbook of linear algebra (L. Hogben, ed.), CRC Press, 2006. (Cited on page 17.)

[HIT85] J. Hale, E. F. Infante, and F.-S. Tsen, *Stability in linear delay equations*, J. Math. Anal. Appl. 105 (1985), 533–555. (Cited on page 132.)

[HJ91] R. Horn and C. Johnson, *Topics in matrix analysis*, Cambridge University Press, Cambridge, UK, 1991. (Cited on pages 17 and 18.)

[HJZ84] D. Hertz, E. Jury, and E. Zeheb, *Simplified analytic stability test for systems with commensurate time delays*, IEE Proc., Part D 131 (1984), 52–56. (Cited on pages 85 and 88.)

[HKP05] M. E. Hochstenbach, T. Košir, and B. Plestenjak, *A Jacobi–Davidson type method for the two-parameter eigenvalue problem*, SIAM J. Matrix Anal. Appl. 26 (2005), no. 2, 477–497. (Cited on page 132.)

[HP03] M. E. Hochstenbach and B. Plestenjak, *Backward error, condition numbers, and pseudospectra for the multiparameter eigenvalue problem*, Linear Algebra Appl. 375 (2003), 63–81. (Cited on page 131.)

[HP05] D. Hinrichsen and A. J. Pritchard, *Mathematical systems theory I*, Springer-Verlag Berlin, 2005. (Cited on pages 153, 156, and 166.)

[HS96] T. Hara and J. Sugie, *Stability region for systems of differential-difference equations*, Funkc. Ekvacioj, Ser. Int. 39 (1996), no. 1, 69–86. (Cited on pages 85 and 87.)

[HS05] P. Hövel and E. Schöll, *Control of unstable steady states by time-delayed feedback methods*, Phys. Rev. E 72 (2005), 046203. (Cited on page 17.)

[HV93] J. Hale and S. M. Verduyn Lunel, *Introduction to functional differential equations*, Springer-Verlag, 1993. (Cited on pages 5, 10, 25, 37, 38, 39, 41, 42, and 73.)

[HV04] C. Huang and S. Vandewalle, *An analysis of delay-dependent stability for ordinary and partial differential equations with fixed and distributed delays*, SIAM J. Sci. Comput. 25 (2004), no. 5, 1608–1632. (Cited on page 79.)

[IK75] O. H. Ibarra and C. E. Kim, *Fast approximation algorithms for the knapsack and sum of subset problems*, J. Assoc. Comput. Mach. 22 (1975), 463–468. (Cited on page 143.)

[IND+03] D. Ivănescu, S.-I. Niculescu, L. Dugard, J.-M. Dion, and E. I. Verriest, *On delay-dependent stability for linear neutral systems*, Automatica 39 (2003), no. 2, 255–261. (Cited on pages 81 and 125.)

[IS02] T. Insperger and G. Stépán, *Stability chart for the delayed mathieu equation*, Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci. 458 (2002), no. 2024, 1989–1998. (Cited on page 38.)

[Jar06a] E. Jarlebring, *Computing the stability region in delay-space of a tds using polynomial eigenproblems*, IFAC Workshop on Time Delay Systems (TDS'06) L'Aquila, Italy, 2006. (Cited on pages 85, 89, and 136.)

[Jar06b] ———, *A quadratic eigenproblem in the analysis of a time delay system*, Proc. Appl. Math. Mech. 6 (2006), 63–66. (Cited on page 127.)

[Jar07] ———, *On critical delays for linear neutral delay systems*, Proc. Europ. Contr. Conf, 2007. (Cited on pages 85 and 113.)

[JD07] E. Jarlebring and T. Damm, *The Lambert W function and the spectrum of some multidimensional time-delay systems*, Automatica 43 (2007), no. 12, 2124–2128. (Cited on page 17.)

[JHC96] D. J. Jeffrey, D. Hare, and R. Corless, *Unwinding the branches of the Lambert W function*, Math. Scientist 21 (1996), 1–7. (Cited on page 17.)

[JS83] N. K. Jain and K. Singhal, *On Kublanovskaya's approach to the solution of the generalized latent value problem for functional lambda-matrices*, SIAM J. Numer. Anal. 20 (1983), 1062–1070. (Cited on page 48.)

[JSH83] N. K. Jain, K. Singhal, and K. Huseyin, *On roots of functional lambda matrices*, Comput. Methods Appl. Mech. Eng. 40 (1983), 277–292. (Cited on page 48.)

[Kam80] E. W. Kamen, *On the relationship between zero criteria for two-variable polynomials and asymptotic stability of delay differential equations*, IEEE Trans. Autom. Control 25 (1980), 983–984. (Cited on page 132.)

[Kam82] ———, *Linear systems with commensurate time delays: Stability and stabilization independent of delay*, IEEE Trans. Autom. Control 27 (1982), 367–375. (Cited on page 132.)

[Kat95] T. Kato, *Perturbation theory for linear operators*, Springer-Verlag, 1995. (Cited on page 147.)

[KB01] W. Kirk and S. Brailey (eds.), *Handbook of metric fixed point theory*, Dordrecht: Kluwer Academic Publishers., 2001. (Cited on pages 146, 163, and 166.)

[KKT06] M. Karow, D. Kressner, and F. Tisseur, *Structured eigenvalue condition numbers*, SIAM J. Matrix Anal. Appl. 28 (2006), no. 4, 1052–1068. (Cited on page 150.)

[KN05] T. Kalmár-Nagy, *A novel method for efficient numerical stability analysis of delay-differential equations*, Proceedings of the America control conference, 2005. (Cited on page 17.)

[Kub70] V. Kublanovskaya, *On an approach to the solution of the generalized latent value problem for λ-matrices*, SIAM J. numer. Anal. 7 (1970), 532–537. (Cited on page 48.)

[Lam91] J. Lambert, *Numerical methods for ordinary differential systems: the initial value problem*, Chichester etc.: John Wiley & Sons, 1991. (Cited on pages 31 and 34.)

[Lan64] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math. 6 (1964), 377–387. (Cited on page 153.)

[Lan02] ———, *Lambda-matrices and vibrating systems*, Mineola, NY: Dover Publications, 2002. (Cited on page 49.)

[LBLK06] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko, *Solving large scale nonlinear eigenvalue problems in next-generation accelerator design*, 2006. (Cited on pages 50, 54, 55, and 162.)

[Lev06] I. Levitskaya, *Stability domain of a linear differential equation with two delays*, Comput. Math. Appl. 51 (2006), no. 1, 153–159. (Cited on page 85.)

[LH69] M. Lee and C. Hsu, *On the τ-decomposition method of stability analysis for retarded dynamical systems*, SIAM J. Control 7 (1969), 242–259. (Cited on page 85.)

[Lim85] T.-C. Lim, *On fixed point stability for set-valued contractive mappings with applications to generalized differential equations*, J. Math. Anal. Appl. 110 (1985), 436–441. (Cited on pages 163, 167, and 168.)

[Lin94] E. Lindelöf, *Sur l'application de la méthode des approximations successives aux équations différentielles ordinaires du premier ordre*, Comptes rendus hebdomadaires des séances de l'Académie des sciences 114 (1894), 454–457. (Cited on page 26.)

[LJH06] X. Li, J. Ji, and C. H. Hansen, *The dynamics of two coupled van der Pol oscillators with delay coupling*, Mech. Research. Comm. 33 (2006), 614–627. (Cited on page 17.)

[LM04] J. Li and Z. Ma, *Stability switches in a class of characteristic equations with delay-dependent parameters*, Nonlinear Anal., Real World Appl. 5 (2004), no. 3, 389–408. (Cited on page 85.)

[Lou01] J. Louisell, *A matrix method for determining the imaginary axis eigenvalues of a delay system*, IEEE Trans. Autom. Control 46 (2001), no. 12, 2008–2012. (Cited on pages 85, 89, 131, 136, 137, and 138.)

[LP05] P. Lancaster and P. Psarrakos, *On the pseudospectra of matrix polynomials*, SIAM J. Matrix Anal. Appl. 27 (2005), no. 1, 115–129. (Cited on page 149.)

[LRW99] X. Li, S. Ruan, and J. Wei, *Stability and bifurcation in delay-differential equations with two delays*, J. Math. Anal. Appl. 236 (1999), no. 2, 254–280. (Cited on page 85.)

[LW05] X. Li and J. Wei, *On the zeros of a fourth degree exponential polynomial with applications to a neural network model with delays*, Chaos Solitons Fractals 26 (2005), no. 2, 519–526. (Cited on page 85.)

[Mah82] J. M. Mahaffy, *A test for stability of linear differential delay equations*, Quarterly of Applied Math 40 (1982), 193–202. (Cited on page 85.)

[Mat07] H. Matsunaga, *Exact stability criteria for delay differential and difference equations*, Applied Mathematics Letters 20 (2007), no. 2, 183–188. (Cited on page 87.)

[MEA07] U. Münz, C. Ebenbauer, and F. Allgöwer, *Stability of networked systems with multiple delays using linear programming*, Proc. 26th American Control Conference, New York, 2007, pp. 5515 – 5520. (Cited on pages 85 and 88.)

[Mee01] K. Meerbergen, *Locking and restarting quadratic eigenvalue solvers*, SIAM J. Sci. Comput. 22 (2001), no. 5, 1814–1839. (Cited on pages 54 and 55.)

[MERD02] W. Michiels, K. Engelborghs, D. Roose, and D. Dochain, *Sensitivity to infinitesimal delays in neutral equations*, SIAM J. Control Optimization 40 (2002), no. 4, 1134–1158. (Cited on pages 73 and 113.)

[MGWN06] W. Michiels, K. Green, T. Wagenknecht, and S.-I. Niculescu, *Pseudospectra and stability radii for analytic matrix functions with application to time-delay systems*, Lin. Alg. Appl. 418 (2006), no. 1, 315–335. (Cited on pages 148 and 163.)

[Min48] N. Minorsky, *Self-excited mechanical oscillations*, J. Appl. Phys. 19 (1948), no. 4, 332–338. (Cited on page 84.)

[MJZ95] J. M. Mahaffy, K. M. Joiner, and P. J. Zak, *A geometric analysis of stability regions for a linear differential equation with two delays*, Int. J. Bifurcation Chaos Appl. Sci. Eng. 5 (1995), no. 3, 779–796. (Cited on page 85.)

[MM06] E. Malakhovski and L. Mirkin, *On stability of second-order quasi-polynomials with a single delay*, Automatica 42 (2006), no. 6, 1041–1047. (Cited on pages 85 and 87.)

[MMMM06a] S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl. 28 (2006), 1029–1051. (Cited on pages 106, 138, 149, 171, and 172.)

[MMMM06b] ———, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl. 28 (2006), 971–1004. (Cited on pages 34, 43, 47, 99, 106, 138, 149, 171, and 172.)

[MN02] C.-I. Morărescu and S.-I. Niculescu, *Old and new in stability analysis of cushing equation: A geometric perspective*, Tech. report, 2002. (Cited on page 94.)

[MN07a] W. Michiels and S.-I. Niculescu, *Characterization of delay-independent stability and delay interference phenomena*, SIAM J. Control Optimization 45 (2007), no. 6, 2138–2155. (Cited on pages 80 and 94.)

[MN07b] ———, *Stability and stabilization of time-delay systems: An eigenvalue-based approach*, Advances in Design and Control 12, SIAM Publications, Philadelphia, 2007. (Cited on pages 6, 10, 25, 73, 74, 83, 93, 113, 114, and 157.)

[MNG06] C.-I. Morărescu, S.-I. Niculescu, and K. Gu, *Remarks on the stability crossing curves of some distributed delay systems*, Proc. of the int. conf. on differential and difference equations and their appl., Melbourne, USA, August, 2005., 2006, pp. 815–823. (Cited on pages 74 and 94.)

[MNG07] ———, *On the geometry of stability regions of smith predictors subject to delay uncertainty*, IMA J. Math. Control Inf. 24 (2007), no. 3, 411–423. (Cited on page 94.)

[Mor06] C.-I. Morărescu, *Qualitative analysis of distributed delay systems: Methodology and algorithms*, Ph.D. thesis, Université de Technologie de Compiègne and Universitatea din Bucureşti, 2006. (Cited on pages 74 and 93.)

[MP99a] P. Mäkilä and J. Partington, *Laguerre and Kautz shift approximations of delay systems*, Int. J. Control 72 (1999), no. 10, 932–946. (Cited on page 45.)

[MP99b] ———, *Shift operator induced approximations of delay systems*, SIAM J. Control Optimization 37 (1999), no. 6, 1897–1912. (Cited on page 45.)

[Muf64] I. Mufti, *A note on the stability of an equation of third order with time lag*, IEEE Trans. Autom. Control 9 (1964), 190–191. (Cited on page 85.)

[MV04] V. Mehrmann and H. Voss, *Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods*, GAMM Mitteilungen 27 (2004), 121–152. (Cited on pages 11, 46, and 47.)

[MV05] W. Michiels and T. Vyhlídal, *An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type*, Automatica 41 (2005), no. 6, 991–998. (Cited on page 125.)

[MVZ+07] W. Michiels, T. Vyhlídal, P. Zítek, H. Nijmeijer, and D. Henrion, *Strong stability of neutral equations with an arbitrary delay dependency structure*, Tech. report, Kath. Univ. Leuven, 2007, submitted to Journal of Differential Equations. (Cited on page 85.)

[Nad69] S. B. Nadler, *Multi-valued contraction mappings*, Pac. J. Math. 30 (1969), 475–488. (Cited on pages 164 and 167.)

[Neu85] A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal. 22 (1985), 914–923. (Cited on pages 12 and 53.)

[NFC05] S.-I. Niculescu, P. Fu, and J. Chen, *Stability switches and reversals of linear systems with commensurate delays: a matrix pencil characterization*, Proc. 16th IFAC Worl Congress, Prague, Czech Republic, 2005. (Cited on pages 85, 89, and 131.)

[NFC06] ———, *On the stability of linear delay-differential algebraic systems: Exact conditions via matrix pencil solutions*, Proceedings of the 45th IEEE Conference on Decision and Control, 2006. (Cited on pages 85, 89, 131, and 137.)

[Nic98] S.-I. Niculescu, *Stability and hyperbolicity of linear systems with delayed state: A matrix-pencil approach*, IMA J. Math. Control Inf. 15 (1998), no. 4, 331–347. (Cited on pages 85, 136, and 137.)

[Nic01a] ———, *Delay effects on stability. a robust control approach*, Springer-Verlag London, 2001. (Cited on pages 5, 74, 76, 78, 83, 108, 123, and 132.)

[Nic01b] ———, *On delay-dependent stability under model transformations of some neutral linear systems*, Int. J. Control 74 (2001), no. 6, 609–617. (Cited on page 83.)

[NKGL06] S.-I. Niculescu, P. S. Kim, K. Gu, and D. Levy, *On the stability crossing boundaries of some delay systems modeling immune dynamics in leukemia*, Proc. 17th Int. Symp. on Math. Theory of Networks and Systems, Kyoto, 2006. (Cited on pages 74 and 87.)

[OS02] N. Olgac and R. Sipahi, *An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems*, IEEE Trans. Autom. Control 47 (2002), no. 5, 793–797. (Cited on pages 85 and 88.)

[OS03] ———, *The direct method for stability analysis of time delayed LTI systems*, Proceedings of the American Control Conference, vol. 1, 2003, pp. 869 – 874. (Cited on page 88.)

[OS04] ———, *A practical method for analyzing the stability of neutral type LTI-time delayed systems*, Automatica 40 (2004), no. 5, 847–853. (Cited on pages 85 and 88.)

[OVS08] N. Olgac, T. Vyhlídal, and R. Sipahi, *A new perspective in the stability assessment of neutral systems with multiple and cross-talking delays*, SIAM J. Control Optim. 47 (2008), no. 1, 327–344. (Cited on page 85.)

[Par91] J. Partington, *Approximation of delay systems by Fourier-Laguerre series*, Automatica 27 (1991), no. 3, 569–572. (Cited on page 45.)

[Par04a] ———, *Linear operators and linear systems.*, Cambridge University Press, 2004. (Cited on pages 45 and 88.)

[Par04b] ———, *Model reduction of delay systems*, pp. 29–32, Princeton university press, 2004, V. Blondel (ed) and A. Megretski (ed). (Cited on page 46.)

[Par04c] ———, *Some frequency-domain approaches to the model reduction of delay systems*, Annual Reviews in Control 28 (2004), no. 1, 65–73. (Cited on pages 45 and 46.)

[PB04] J. Partington and C. Bonnet, H_∞ *and BIBO stabilization of delay systems of neutral type*, Syst. Control Lett. 52 (2004), 283–288. (Cited on page 45.)

[Pee06] M. Peet, *Stability and control of functional differential equations*, Ph.D. thesis, Stanford University, 2006. (Cited on page 79.)

[PFF+06] K. Patanarapeelert, T. Frank, R. Friedrich, P. Beek, and I. Tang, *Theoretical analysis of destabilization resonances in time-delayed stochastic second-order dynamical systems and some implications for human motor control*, Phys. Rev. E 73 (2006), 021901. (Cited on page 74.)

[Pio07] M. Piotrowska, *A remark on the ODE with two discrete delays*, J. Math. Anal. Appl. 329 (2007), no. 1, 664–676. (Cited on page 85.)

[Pit05] A. Pitcher, *On a numerical method for solving homogeneous partial delay differential equations with constant coefficients and time independent delays*, Tech. report, University of Western Ontario, 2005. (Cited on page 17.)

[Pli05] E. Plischke, *Transient effects of linear dynamical systems*, Ph.D. thesis, Universität Bremen, 2005. (Cited on pages 73 and 74.)

[PM05] J. Partington and P. Mäkilä, *Rational approximation of distributed-delay controllers*, Int. J. Control 78 (2005), no. 16, 1295–1301. (Cited on page 45.)

[PW79] G. Peters and J. Wilkinson, *Inverse iterations, ill-conditioned equations and Newton's method*, SIAM Rev. 21 (1979), 339–360. (Cited on page 52.)

[Rei05] T. Reis, *Model reduction for a class of PDAE systems*, Proc. of GAMM Annual meeting, vol. 5, 2005, doi:10.1002/pamm.200510068, pp. 179–180. (Cited on page 46.)

[Rek80] Z. Rekasius, *A stability test for systems with delays*, Proc. of joint Autom. Contr. Conf San Francisco, 1980, pp. TP9–A. (Cited on pages 85 and 88.)

[RKDD04] S. Rodrìguez, V. Kharitonov, J.-M. Dion, and L. Dugard, *Robust stability of neutral systems: A Lyapunov-Krasovskii constructive approach*, Int. J. Robust Nonlinear Control 14 (2004), no. 16, 1345–1358. (Cited on page 125.)

[Rog64] E. Rogers, *A minimax theory for overdamped systems*, Arch. Ration. Mech. Anal. 16 (1964), 89–96. (Cited on page 50.)

[RR00] H. Radjavi and P. Rosenthal, *Simlutaneuos triangularization*, Springer Verlag, 2000. (Cited on page 20.)

[Rud76] W. Rudin, *Principles of mathematical analysis. 3rd ed*, McGraw-Hill, 1976. (Cited on page 153.)

[Ruh73] A. Ruhe, *Algorithms for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal. 10 (1973), 674–689. (Cited on pages 11, 47, 49, 50, and 161.)

[Rum06] S. M. Rump, *Eigenvalues, pseudospectrum and structured perturbations*, Linear Algebra Appl. 413 (2006), 567–593. (Cited on page 150.)

[RV07] R. Rand and A. Verdugo, *Hopf bifurcation formula for first order differential-delay equations*, Communications in Nonlinear Science and Numerical Simulation 12 (2007), no. 6, 859–864. (Cited on pages 84 and 85.)

[SBFvdV96] G. L. Sleijpen, A. G. Booten, D. R. Fokkema, and H. A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT 36 (1996), no. 3, 595–633. (Cited on pages 34, 54, 100, 107, and 172.)

[SLHT07] C. Sun, Y. Lin, M. Han, and S. Tang, *Analysis for a special first order characteristic equation with delay dependent parameters*, Chaos, Solitons & Fractals 33 (2007), no. 2, 388–395. (Cited on page 85.)

[SM06] H. Shinozaki and T. Mori, *Robust stability analysis of linear time-delay systems by Lambert W function: Some extreme point results*, Automatica 42 (2006), no. 10, 1791–1799. (Cited on page 17.)

[SO05] R. Sipahi and N. Olgac, *Complete stability robustness of third-order LTI multiple time-delay systems*, Automatica 41 (2005), no. 8, 1413–1422. (Cited on pages 74, 85, and 88.)

[SO06a] ———, *Complete stability analysis of neutral-type first order two-time-delay systems with cross-talking delays*, SIAM J. Control Optimization 45 (2006), no. 3, 957–971. (Cited on pages 85 and 88.)

[SO06b] ———, *Stability robustness of retarded LTI systems with single delay and exhaustive determination of their imaginary spectra*, SIAM J. Control Optim. 45 (2006), no. 5, 1680–1696. (Cited on page 85.)

[SO06c] ———, *A unique methodology for the stability robustness of multiple time delay systems*, Syst. Control Lett. 55 (2006), no. 10, 819–825. (Cited on page 88.)

[Son98] E. Sontag, *Mathematical control theory, deterministic finite dimensional systems*, 2nd ed., Springer-Verlag, New York, 1998. (Cited on page 21.)

[SR02] K. V. Singh and Y. M. Ram, *Transcendental eigenvalue problem and its applications*, AIAA Journal 40 (2002), no. 7, 1402–1407. (Cited on page 49.)

[SS90] G. Stewart and J. Sun, *Matrix perturbation theory*, Academic Press, 1990. (Cited on pages 147, 153, 164, 165, and 166.)

[SS06] H. Schwetlick and K. Schreiber, *A primal-dual Jacobi-Davidson-like method for nonlinear eigenvalue problems*, Tech. Report ZIH-IR-0613, pp. 1–20, Techn. Univ. Dresden, Zentrum für Informationsdienste und Hochleistungsrechnen, 2006. (Cited on page 54.)

[Sté89] G. Stépán, *Retarded dynamical systems: stability and characteristic functions*, John Wiley & Son, New York, 1989. (Cited on pages 79 and 85.)

[SvdV96] G. L. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. 17 (1996), no. 2, 401–425. (Cited on page 58.)

[TE05] L. N. Trefethen and M. Embree, *Spectra and pseudospectra. the behavior of nonnormal matrices and operators*, Princeton University Press, 2005. (Cited on pages 147 and 149.)

[TH01] F. Tisseur and N. J. Higham, *Structured pseudospectra for polynomial eigenvalue problems, with applications*, SIAM J. Matrix Anal. Appl. 23 (2001), no. 1, 187–208. (Cited on pages 148 and 149.)

[Tho81] A. Thowsen, *An analytic stability test for class of time-delay systems*, IEEE Trans. Autom. Control 26 (1981), no. 3, 735–736. (Cited on pages 85 and 88.)

[Thu78] G. Thurston, *Roots of lambda matrices*, ASME J. Appl. Mech. 45 (1978), no. 4, 859–863. (Cited on page 50.)

[TM01] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev. 43 (2001), no. 2, 235–286. (Cited on page 98.)

[Tre00] L. N. Trefethen, *Spectral methods in MATLAB*, SIAM Publications, Philadelphia, 2000. (Cited on pages 37, 43, and 44.)

[TT94] K.-C. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math. 68 (1994), no. 3, 403–425. (Cited on page 149.)

[TÖ96] O. Toker and H. Özbay, *Complexity issues in robust stability of linear delay-differential systems*, Math. Control Signals Syst. 9 (1996), no. 4, 386–400. (Cited on pages 71, 90, 140, and 141.)

[Ung50] H. Unger, *Nichtlineare Behandlung von Eigenwertaufgaben*, Z. Angew. Math. Mech. 30 (1950), 281–282, English translation: http://www.math.tu-dresden.de/ schwetli/Unger.html. (Cited on page 52.)

[VGR04] K. Verheyden, K. Green, and D. Roose, *Numerical stability analysis of a large-scale delay system modeling a lateral semiconductor laser subject to optical feedback*, Phys. Rev. E 69 (2004), 036702. (Cited on pages 13 and 67.)

[Vos03] H. Voss, *A maxmin principle for nonlinear eigenvalue problems with application to a rational spectral problem in fluid-solid vibration*, Appl. Math., Praha 48 (2003), no. 6, 607–622. (Cited on pages 11 and 51.)

[Vos04a] ———, *An Arnoldi method for nonlinear eigenvalue problems*, BIT numerical mathematics 44 (2004), 387 – 401. (Cited on pages 12, 13, 51, 54, 55, 64, and 66.)

[Vos04b] ———, *Numerical methods for sparse nonlinear eigenvalue problems*, Proc. XVth Summer School on Software and Algorithms of Numerical Mathematics, Hejnice, Czech Republic, 2004, Report 70. Arbeitsbereich Mathematik, TU Hamburg-Harburg. (Cited on pages 49, 50, 51, and 162.)

[Vos06a] ———, *Iterative projection methods for computing relevant energy states of a quantum dot*, J. Comput. Phys. 217 (2006), no. 2, 824–833. (Cited on page 13.)

[Vos06b] ———, *Projection methods for nonlinear sparse eigenvalue problems*, Annals of the European Academy of Sciences (2006), 152–183. (Cited on page 55.)

[Vos07] ———, *A Jacobi-Davidson method for nonlinear and nonsymmetric eigenproblems*, Computers & Structures 85 (2007), no. 17, 1284–1292. (Cited on page 58.)

[VW82] H. Voss and B. Werner, *A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems*, Math. Methods Appl. Sci. 4 (1982), 415–424. (Cited on page 11.)

[Vyh03] T. Vyhlídal, *Analysis and synthesis of time delay system spectrum*, Ph.D. thesis, Czech Technical University Prague, 2003. (Cited on page 15.)

[VZ06] T. Vyhlídal and P. Zítek, *Mapping the spectrum of a retarded time-delay system utilizing root distribution features*, Proc of IFAC Workshop on Time-Delay Systems, L'Aquila, 2006, TuAd. (Cited on page 15.)

[WC05] P. Wahi and A. Chatterjee, *Galerkin projections for delay differential equations*, J. Dyn. Syst.-T. ASME 127 (2005), 80–87. (Cited on page 17.)

[WD07] R. Warburton and S. Disney, *Order and inventory variance amplification: The equivalence of discrete and continuous time analyses*, Int. J. Production Economics 110 (2007), 128–137. (Cited on page 17.)

[Wer70] B. Werner, *Das Spektrum von Operatorenscharen mit verallgemeinerten Rayleighquotienten*, Ph.D. thesis, Fachbereich Mathematik, Universität Hamburg, 1970. (Cited on page 50.)

[Wer72] ———, *Min-Max-Prinzipien für nichtlinear auftretende Eigenwerte bei klassischen und natürlichen Randeigenwertproblemen*, Computing 9 (1972), 189–201. (Cited on page 11.)

[WM87] K. Walton and J. Marshall, *Direct method for TDS stability analysis*, IEE Proc., Part D 134 (1987), 101–107. (Cited on pages 85 and 88.)

[WMG08] T. Wagenknecht, W. Michiels, and K. Green, *Structured pseudospectra for nonlinear eigenvalue problems*, J. Comput. Appl. Math. 211 (2008), 245–259, doi: 10.1016/j.cam.2006.12.00. (Cited on page 148.)

[Wu96] J. Wu, *Theory and applications of partial functional differential equations*, Applied Mathematical Sciences. 119. New York, NY: Springer., 1996. (Cited on pages 60 and 110.)

[Yan83] W. H. Yang, *A method for eigenvalues of sparse lambda-matrices*, Int. J. Numer. Methods Eng. 19 (1983), 943–948. (Cited on page 49.)

[YU06] S. Yi and A. G. Ulsoy, *Solution of a system of linear delay differential equations using the matrix lambert function*, Proceedings of American Control Conference, 2006, pp. 2433–2438. (Cited on page 17.)

[YUN06] S. Yi, A. G. Ulsoy, and P. Nelson, *Analysis of systems of linear delay differential equations using the matrix Lambert function and the Laplace transformation*, Tech. report, Univ. Michigan, 2006, submitted to Automatica. (Cited on page 17.)

[Zaf07] N. Zafer, *Discussion: "analysis of a system of linear delay differential equations"*, J. Dyn. Sys. and Contr. 129 (2007), 121–122. (Cited on page 17.)

[Zho06] Q.-C. Zhong, *Robust control of time-delay systems*, Springer Verlag, 2006. (Cited on pages 3 and 5.)

Index

Address:
Institut *Computational Mathematics*
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig
Germany

office: +49(0)5313917541
e-mail: e.jarlebring@tu-bs.de

Elias Jarlebring

Curriculum Vitae

- Name: Elias Jarlebring
- Birth: February 3, 1978, Vännäs, Sweden
- Nationality: Swedish

Academic Background

October 2004 -
- Research associate (wissenschaftlicher Mitarbeiter) at the *technical university of Braunschweig, Institut computational mathematics.*

July 1998 - September 2003
- September 24th, 2003, Teknisk fysik, Master of Science, *Royal Institute of Technology* (KTH), Stockholm, Sweden, with two time-periods spent abroad as integral parts of the studies (see below).

Sept 2002 - March 2003
- Master's thesis [7] performed as an exchange student at *Technische Universität Hamburg-Harburg* (TUHH) investigating the topic of *Krylov methods for nonlinear eigenvalue problems.* Further results were published in [2] by the author and H. Voss. The examiner of the thesis was Axel Ruhe.

Aug 2000 - May 2001
- International student at *Trinity College Dublin* (TCD), Ireland.
- Swedish compulsory school with one year spent at Donegal high-school, USA and gymnasial studies (3 years) at Forsmark computer-science gymnasium.

Teaching Experience

Oct 2004 -
- Teaching numerical analysis as a part of the PhD position at TUBS.

Oct 2002 - March 2003
- Teacher for the recitation classes in linear algebra at TUHH.

Oct 1999 - April 2000
- Teacher and laboratory assistant in computer programming at KTH.

Other Professional Experience

March 2000 - July 2000
May 2003 - Oct 2004
- Computer consultant at Racasse AB (Stockholm).

June 2000 - Aug 2001
- Position as a *werkstudent* at the department of engineering and computer development at Siemens AG, Munich, Germany.

Language Skills

- Swedish - fluent (native language)
- English - fluent
- German - well spoken

Refereeing Activities

- Automatica, Linear Algebra Appl., J. Comp. Appl. Math., AMS Mathematical Reviews

List of Publications

Journal Articles

[1] • E. Jarlebring and T. Damm.
The Lambert W function and the spectrum of some multidimensional time-delay systems.
Automatica, 43(12):2124–2128, 2007.

[2] • E. Jarlebring and H. Voss.
Rational Krylov for nonlinear eigenproblems, an iterative projection method.
Applications of Mathematics, 50(6):534–554, 2005.

[3] • E. Jarlebring.
Critical delays and polynomial eigenvalue problems, 2008, doi:10.1016/j.cam.2008.05.004
accepted for publication in *Journal of Computational and Applied Mathematics*,

Refereed conference proceedings

[4] • E. Jarlebring.
Computing the stability region in delay-space of a tds using polynomial eigenproblems.
In *IFAC Workshop on Time Delay Systems (TDS'06) L'Aquila, Italy*, 2006.

[5] • E. Jarlebring.
A quadratic eigenproblem in the analysis of a time delay system.
Proc. Appl. Math. Mech., 6:63–66, 2006.

[6] • E. Jarlebring.
On critical delays for linear neutral delay systems.
In *Proc. Europ. Contr. Conf*, 2007.

Other publications

[7] • E. Jarlebring.
Krylov Methods for Nonlinear Eigenvalue Problems.
Master's thesis, Royal Institute of Technology, Stockholm, 2003.