Solving delay eigenvalue problems by rational approximations

Bartalus, G.

Award date:
2009

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Solving Delay Eigenvalue Problems by Rational Approximations

Industrial and Applied Mathematics Master's Thesis

Gaspar Bartalus
0641392

Supervisor: Michiel E. Hochstenbach

Eindhoven, August 2009

Contents

1 Introduction **7**

2 Polynomial and rational approximations **11**

2.1 Rational approximations based on Taylor expansion 11

2.1.1 Padé approximations 11

2.1.2 Alternative Padé approximations for e^{-z} 13

2.2 Rational interpolations 16

2.2.1 Chebyshev rational interpolation [25] 17

2.2.2 Legendre approximation [25] 17

2.3 Polynomial approximations 17

2.3.1 Lagrange interpolation 17

2.3.2 The Remez method for minimax approximation 19

2.3.3 The Remez method for rational approximation 22

2.4 Overdetermined rational approximation on a complex domain 24

2.5 Best approximation on the unit disk 27

2.5.1 The CF method 28

2.5.2 Numerical computation of r^* 29

2.5.3 Numerical example 30

2.6 Rational Chebyshev approximation in the complex plane 31

3 Delay Eigenvalue Problems **35**

3.1 Introduction 35

3.2 "Exact" eigenvalues 37

3.2.1 PDE formulation of a DDE 37

3.2.2 Discretizing the PDE 38

3.2.3 Reference eigenvalues 40

3.2.4 Solution operator discretization 41

3.3 Newton-type method 43

3.4 Galerkin and minimal residual method 43

3.4.1 Galerkin method 44

3.4.2 Minimum residual method 45

3.5 Approximate eigenproblems 47

3.5.1 Linearization of polynomial eigenproblems [12] 47

3.5.2 REP by Padé approximations for $e^{-\tau\lambda}$ 48

3.5.3 REP by Padé approximations for $\ln \mu$ 52

3.5.4 REP by Chebyshev and Legendre interpolation 53

3.5.5 PEP by minimax approximation 53

3.5.6 PEP by Lagrange interpolation 55

3.5.7 REP by rational minimax approximation 55

3.5.8 REP by least squares approximation on complex box 55

3.5.9 REP by best rational approximation on unit disk 57

3.5.10 REP by best rational approximation on random complex regions . . 57

4 Perturbation theory **59**

4.1 Introduction 59

4.2 Preliminaries 60

4.3 Backward error 60

4.4 Condition numbers 62
4.4.1 Eigenvalue condition number 63
4.5 Pseudospectra 64

5 Conclusions and Outlook **67**

Chapter 1

Introduction

Many physical problems 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 modeling of some problems, however, might require an additional term in the differential equation depending on the state at some previous time-point. These types of differential equations are known as *delay-differential equations* (DDEs), and have an extensive literature due to [1, 7]. Generally, a DDE has the form

$$\dot{x}(t) = A_0 x(t) + \sum_{k=1}^{m} A_k x(t - \tau_k),$$

that is, the derivative of the state is a linear combination of the current state and some previous states. In this thesis we mostly consider the delay differential equation with single-delay, i.e.,

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau), \qquad t > 0,$$

where $A_0, A_1 \in \mathbb{R}^{n \times n}$ and $\tau > 0$.

This generalization of the ODE is important, as it allows the mathematical analysis of models with delays. For instance, the electronic signal of the control of a robot takes some time to go from the controller to the robot arm. Thus, if the controllers of the wing-rudders of an airplane are located in the cockpit (which fortunately is the case in many situations), the controllers can only control the rudders with a certain delay. Furthermore, one of the most critical application of such models with delay is in the field of nuclear reactors. The temperature of the inner part of a 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 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.*

Hot shower problem:

In order to illustrate such a model with delay we shall present in the following the so-called *hot shower problem.* 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. Hence, consider a shower with the following physical parameters: let the length of the shower hose, i.e., the distance from the shower tap to the shower head, be denoted by l; let v be the speed of the water in the hose, taken as constant (stationary flow). The state $x(t)$ is the temperature difference from the optimum, i.e., the difference between the (human dependent) optimal temperature and the present temperature at the shower tap. Moreover, we model the the human being as a linear controller with sensitivity $\alpha > 0$. That is, the change of the controller 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 parameters is given in Figure 1.1.

As indicated by Figure 1.1, the stability conditions are an important issue in the field of DDEs. For an elaborate study of these conditions the reader is referred to [12, Chapter 3.]. As a matter of fact, the DDE (1.1) is stable if and only if

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

Therefore, the reasons for unstable showers are long shower hoses, low water pressure and/or sensitive human beings.

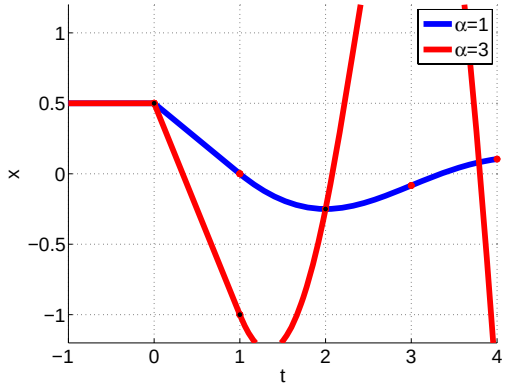

Figure 1.1: Solutions of the hot shower problem.

Overview: The main problem of solving delay eigenvalue problems is approached in Chapter 3. The polynomial and rational approximation techniques used for this are presented in Chapter 2. Chapter 4 is about perturbation results for the delay eigenvalue problem, while Chapter 5 concludes the current thesis.

Chapter 2

Polynomial and rational approximations

In this chapter we review several polynomial and rational approximation methods for the exponential function

$$f(z) = e^{-z}.$$

Most of the described techniques are commonly used and have already an extensive literature. We merely present them as expectedly relevant techniques for dealing with the problems in Chapter 3. The list of methods is, of course, far from being exhaustive in the field. For a deep understanding of polynomial and rational approximation methods, and approximation theory in general, the reader is referred to [14, 4].

2.1 Rational approximations based on Taylor expansion

2.1.1 Padé approximations

Approximants derived by expanding a function $f(z)$ as a ratio of two polynomials are called Padé-approximants. These are usually superior to the Taylor series when the functions contain poles, because the use of rational functions allows them to be well represented. Let $R_{n,m}(z)$ be the quotient approximating $f(z)$:

$$R_{n,m}(z) = \frac{P_n(z)}{Q_m(z)},$$

where

$$P_n(z) = p_0 + p_1 z + \cdots + p_n z^n,$$

$$Q_m(z) = 1 + q_1 z + \cdots + q_m z^m.$$

According to the method of Padé, these two polynomials are constructed, so that at given $z = z_0$ the functions $f(z)$ and $R_{n,m}(z)$, and their derivatives up to order $n+m$ are equal. One can take $z_0 = 0$, and require the function and its derivative to be continuous at this point. We note at this point, that the Padé-approximant $R_{n,0}(z)$ corresponds to the Maclaurin series. Moreover, $q_0 = 1$ is chosen for reasons of uniqueness of the approximation.

Hence, the rational function $R_{n,m}(z)$ has $n+m+1$ unknown coefficients. Assume that $f(z)$ has the Maclaurin expansion

$$f(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_k z^k + \ldots,$$

where

$$a_k = \frac{f^{(k)}(0)}{k!}.$$

Then the coefficients can be found by setting

$$Q_m(z) f(z) - P_n(z) = 0,$$

This will lead to a system of $n+m+1$ linear equations:

$$\begin{aligned}
a_0 - p_0 &= 0 \\
q_1 a_0 + a_1 - p_1 &= 0 \\
q_2 a_0 + q_1 a_1 + a_2 - p_2 &= 0 \\
&\vdots \\
q_m a_{n-m} + q_{m-1} a_{n-m+1} + \cdots + a_n - p_n &= 0
\end{aligned}$$

$$\begin{aligned}
q_m a_{n-m+1} + q_{m-1} a_{n-m+2} + \cdots + q_1 a_n + a_{n+1} &= 0 \\
q_m a_{n-m+2} + q_{m-1} a_{n-m+3} + \cdots + q_1 a_{n+1} + a_{n+2} &= 0 \\
&\vdots \\
q_m a_n + q_{m-1} a_{n+1} + \cdots + q_1 a_{n+m-1} + a_{n+m} &= 0
\end{aligned}$$

which can be easily solved.

Below one can find some Padé approximations for e^{-z} that will be used in the upcoming paragraphs:

$$
\begin{aligned}
R_{1,1}(z) &= \frac{1-\frac{1}{2}z}{1+\frac{1}{2}z} \\
R_{2,1}(z) &= \frac{1-\frac{2}{3}z+\frac{1}{6}z^2}{1+\frac{1}{3}z} \\
R_{2,2}(z) &= \frac{1-\frac{1}{2}z+\frac{1}{12}z^2}{1+\frac{1}{2}z+\frac{1}{12}z^2} \\
R_{4,4}(z) &= \frac{1-\frac{1}{2}z+\frac{3}{28}z^2-\frac{1}{84}z^3+\frac{1}{1680}z^4}{1+\frac{1}{2}z+\frac{3}{28}z^2+\frac{1}{84}z^3+\frac{1}{1680}z^4}
\end{aligned}
$$

An interesting observation regarding these approximations is the following. Let $p(z)$ and $q(z)$ be the two unique polynomials of given order, such that

$$e^{-z} \approx \frac{p(z)}{q(z)}.$$

Thus

$$e^{z} \approx \frac{q(z)}{p(z)} \qquad \text{and} \qquad e^{-z} \approx \frac{q(-z)}{p(-z)}.$$

Since $p(z)$ and $q(z)$ are unique one can conclude that, if $\deg(p) = \deg(q) = m$,

$$p(z) = q(-z), \tag{2.1}$$

and therefore the $(m,m)-$Padé approximation for e^{-z} is of the form

$$e^{-z} \approx \frac{p(z)}{p(-z)}.$$

Figure 2.1 shows the error curve of two different Padé approximations of e^{-z}.

2.1.2 Alternative Padé approximations for e^{-z}

Consider a Padé approximation of the form

$$e^{-z} \approx \frac{p(z)}{q(z)}.$$

Based on [16], an alternative rational approximation can easily be derived as follows:

$$e^{-z} = \left(e^{-\frac{z}{n}}\right)^n \approx \frac{p\left(\frac{z}{n}\right)^n}{q\left(\frac{z}{n}\right)^n}. \tag{2.2}$$

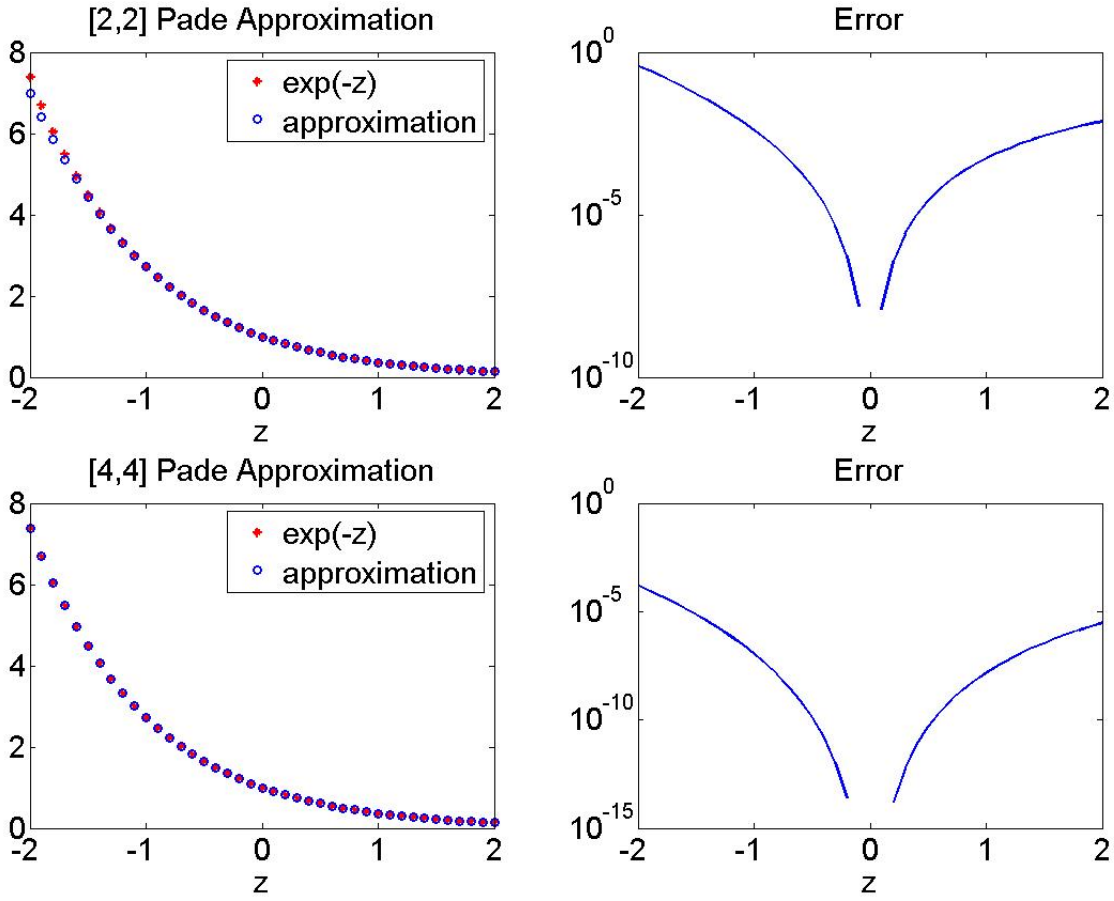

Figure 2.1: Padé approximations of e^{-z} on $[-2, 2]$.

These type of approximations are called shift-based (Padé) approximations [16], and the gain from the use of scaling is that we broaden the domain of approximation. That is, by (2.2), we attempt to improve the quality of the standard Padé approximation further from the origin. For a full description of such approximations the reader is referred to [16]. Note that for $n = 1$ we get the standard Padé approximation.

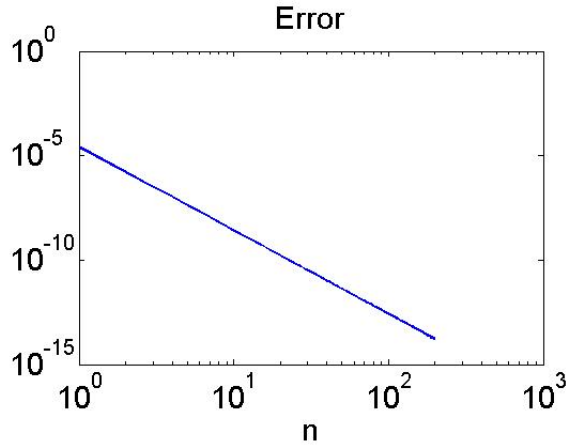

Figure 2.2: Error for shifted Padé approximations of e^{-z}, calculated at $z = 0.5$.

Effectiveness of method

In this section we compare the approximations presented earlier. More explicitly, we make comparisons between approximations of the same degree. In order to avoid confusions we shall introduce the notation $(m, n)^k$-*Padé* for the approximations of the type

$$e^{-z} \approx \left(R_{m,n}\left(\frac{z}{k}\right)\right)^k = \left(\frac{p_m(z/k)}{q_n(z/k)}\right)^k.$$

Hence the comparisons are made between $(2, 2)$- and $(2, 2)^1$-Padé, $(4, 4)$- and $(2, 2)^2$-Padé, $(6, 6)$- and $(2, 2)^3$-Padé, etc. The maximum absolute residuals are then plotted against k for different intervals in Figure 2.3. The results indicate that around the origin the standard Padé approximants are always faster than the shifted ones, but as soon as we move further away from it, the shifted approximations are starting to give better results. What happens in fact is that the standard Padé approximants are starting to get more and more inaccurate as we move away from the origin. Note moreover, that the approximation errors are considerably larger in the furthest interval.

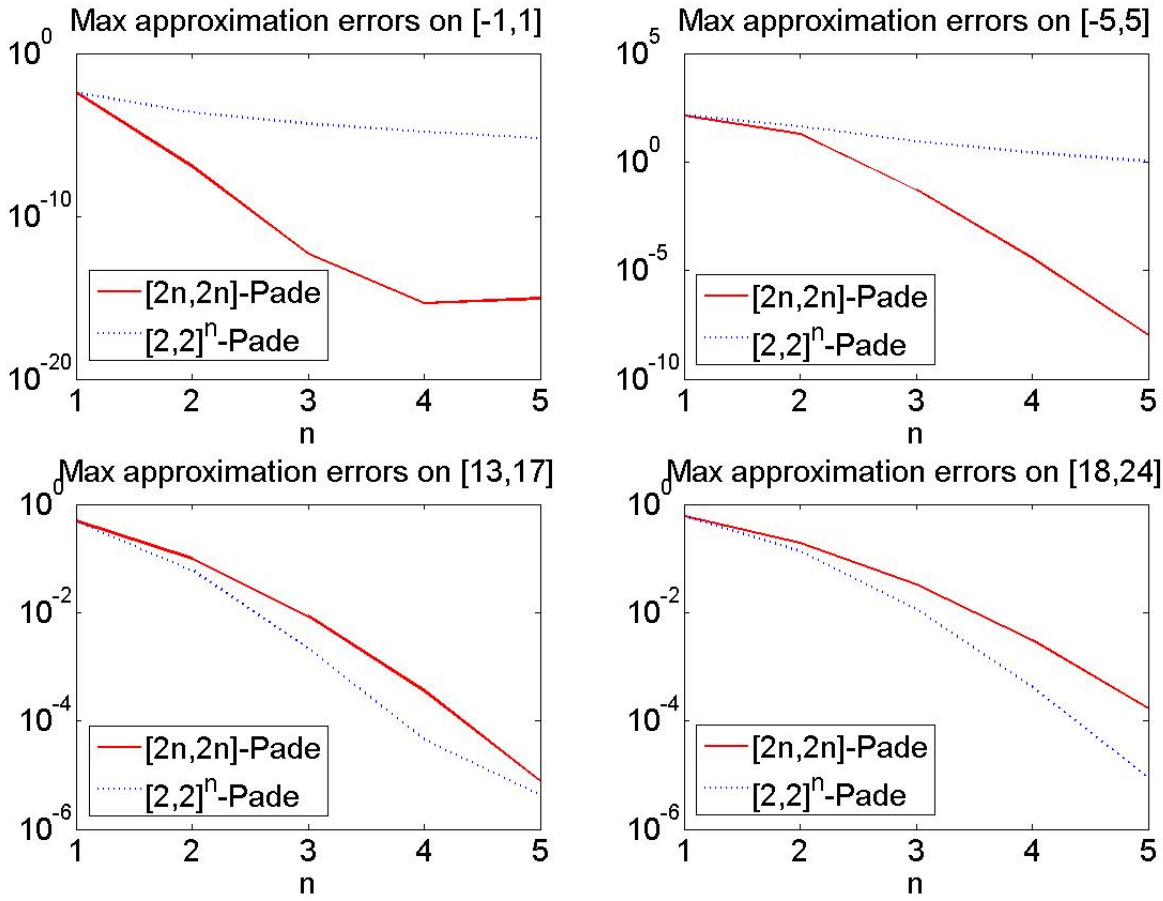

Figure 2.3: Approximations of e^{-z}.

2.2 Rational interpolations

A *rational interpolation* of the function $f(x)$ is defined as the task of determining a rational function $r_{n,m}(x) = p_n(x)/q_m(x)$ such that

$$r_{n,m}(x_j) = f(x_j), \qquad 0 \le j \le n+m+1, \tag{2.3}$$

for certain points x_j. Denoting

$$p_n(x) = p_0 + p_1 x + \cdots + p_n x^n$$

and

$$q_m(x) = 1 + q_1 x + \cdots + q_m x^m,$$

(2.3) reduces to a set of $n+m+1$ linear equations for $n+m+1$ unknown coefficients, which can be easily solved by standard techniques.

According to the choice of $\{x_j\}$ we distinguish the following rational interpolation methods:

2.2.1 Chebyshev rational interpolation [25]

Consider the grid formed by the Chebyshev extreme points

$$x_j = \cos\frac{j\pi}{N}, \qquad j = 0, 1, \ldots, N.$$

The Chebyshev rational interpolation method makes use of these points in order to approximate the function $f(x)$. The result of the interpolation for $f(x) = e^{-x}$ in $[-1, 1]$ is shown in Figure 2.4.

2.2.2 Legendre approximation [25]

Using $x_j = j$th extremum of the Legendre polynomial P_N of degree N, for $j = 0, 1, \ldots, N$ (see [25] for details), we arrive at the so-called Legendre rational interpolation method. An application for $f(x) = e^{-x}$ is shown in Figure 2.4.

The numerical results presented in Figure 2.4 show that by taking the same number of points on the same interval, the Chebyshev and the Legendre rational interpolations yield fairly similar results.

2.3 Polynomial approximations

In this section we shall discuss some methods of approximating e^{-x} by a polynomial function.

2.3.1 Lagrange interpolation

For a given set of $k + 1$ data points

$$(x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k),$$

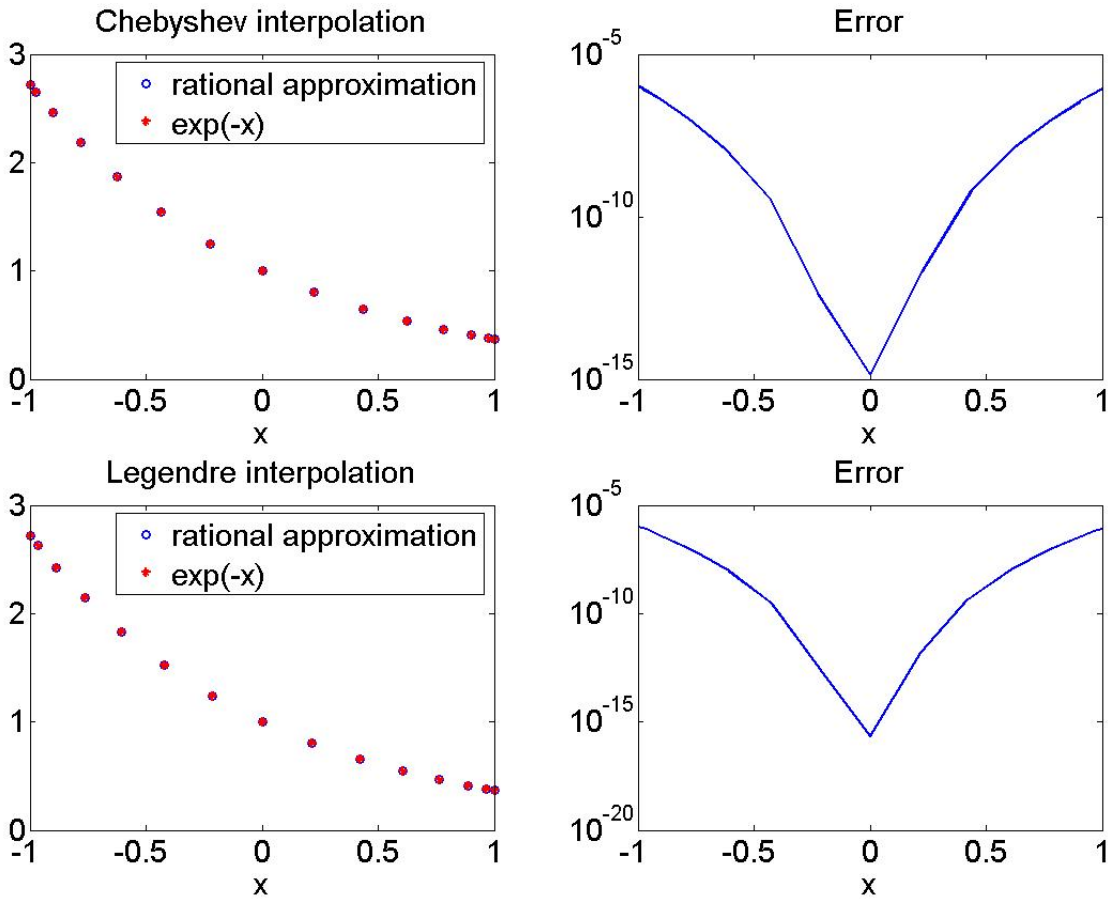

Figure 2.4: An $[14, 14]$ - Chebyshev and Legendre rational interpolation in $[-1, 1]$, with maximum residuals: $9.8 \cdot 10^{-7}$ and $9.7 \cdot 10^{-7}$.

the Lagrange interpolation polynomial is a linear combination

$$L(x) := \sum_{j=0}^{k} y_j l_j(x)$$

of Lagrange basis polynomials

$$l_j(x) := \prod_{i \neq j} \frac{x - x_i}{x_j - x_i}.$$

Choosing $y_j = e^{-x_j}$ for some $x_0, x_1, \ldots, x_k$, one can use the Lagrange interpolation for approximating the function $f(x) = e^{-x}$. Figure 2.5 shows the approximation error for an interpolation of order 28 on the interval $[-1, 1]$. Since we have used the same number of interpolation points on the same interval $[-1, 1]$ as for the previously discussed rational interpolations, one can make a fair comparison between these approximation techniques. The first observation, as expected for a polynomial approximation, is that the error for the Lagrange interpolation has an oscillatory behavior. Moreover, the maximum error on the interval is significantly larger than the maximum error for the Chebyshev or Legendre interpolation.

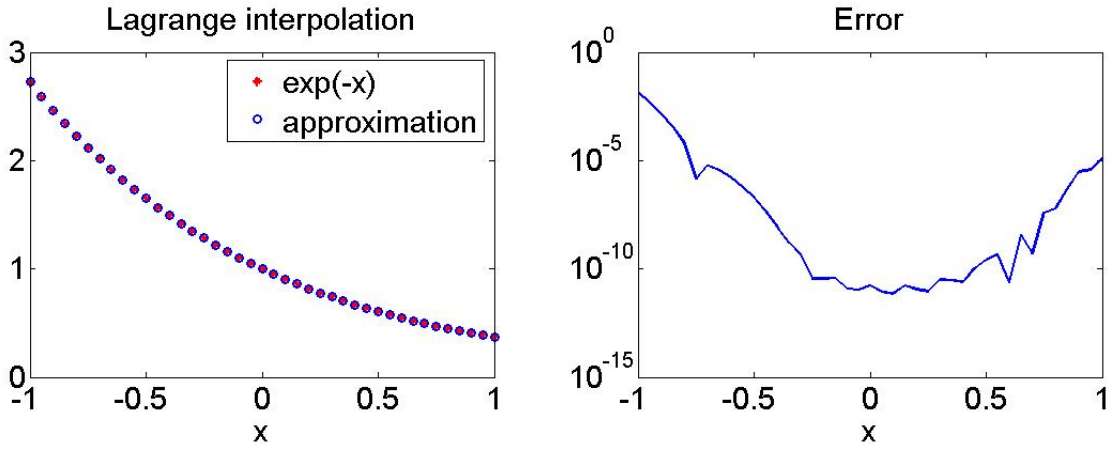

Figure 2.5: Lagrange interpolation of order 28 on $[-1, 1]$. Maximal error $1.3 \cdot 10^{-2}$.

2.3.2 The Remez method for minimax approximation

In this section we wish to find the "best" polynomial approximation $p(x)$ to a given function $f(x)$ on a certain interval $[a, b]$, where "best" is defined to be the approximation that has

the least deviation from $f(x)$. To measure the deviation one can introduce the maximum absolute error of the approximation of $f(x)$ by $p(x)$ on the interval $[a, b]$:

$$E(p) = \max_{a \le x \le b} |f(x) - p(x)|.$$

The minimax approximation is based on finding the polynomial with degree $\le n$ for which the above error is minimized, i.e.

$$\rho_n(f) = \min_{\deg(p) \le n} E(p).$$

The number $\rho_n(f)$ will be the smallest possible absolute error, or minimax error, when approximating $f(x)$ by polynomials of degree at most n. The basic result regarding minimax polynomials (or best approximation polynomials in the sense of Chebyshev) is the classical theorem of Chebyshev, stated here without proof.

Theorem 2.1 (Chebyshev's minimax condition)**.** *A continuous function $f : [a, b] \to \mathbb{R}$ has a unique minimax polynomial of degree $\le n$, such that there exists $\epsilon = \pm 1$ and $n + 2$ points $a \le x_0 < x_1 < \cdots < x_{n+1} \le b$ satisfying*

$$f(x_i) - p(x_i) = \epsilon(-1)^i E(p), \qquad i = 0, 1, \ldots, n + 1.$$

In words, a polynomial $p(x)$ is the minimax polynomial of degree $\le n$ of $f(x)$ if and only if the error function $f(x) - p(x)$ reaches its maximal absolute value $n + 2$ times, with alternating signs.

Hence, knowing the locations $x_0, x_1, \ldots, x_{n+1}$ of the extrema of the error function, one can write up a system of $n + 2$ linear equations

$$p(x_i) + (-1)^i E = f(x_i),$$

with $n + 2$ unknowns, being the polynomial coefficients $c_0, c_1, \ldots, c_n$ and the maximum absolute error E.

In order to find the location of the extrema of the error function, and consequently determine the minimax polynomial for $f(x)$ one can make use of the *Remez method* [18], named after the Russian mathematician E. Ya. Remez. The Remez method is an iterative algorithm. We start with an arbitrary set of $n + 2$ points in the given interval.

Step 1:

The first step in the Remez method, given our current set of $n+2$ control points x_i, is to solve the linear system

$$c_0 + c_1 x_i + \cdots + c_n x_i^n + (-1)^i E = f(x_i),$$

to obtain the error term and the coefficients of the polynomial $p(x)$.

This gives us an approximation to $f(x)$ that has the same error E at each of the control points, and whose error function alternates in sign at the control points. However, this is not necessarily the minimax solution: since the control points may not be at the extrema of the error function. Therefore the minimax condition might still not be met. This takes us to the second step.

Step 2:

The second step of the Remez method seeks a new set of $n+2$ points, that approaches the $n+2$ points of the minimax condition. This is called the exchange step. We start by noting that the error alternates in sign at the control points of the first step, therefore the error function has $n+1$ roots, one in each of the intervals $[x_0, x_1]$, $[x_1, x_2]$,..., $[x_n, x_{n+1}]$. We denote these roots by $z_0, z_1, \ldots, z_n$, and calculate them by standard root finding techniques such as the method of chords or bisection [25]. Once they are found, we know that n extrema are bracketed between each pair of roots, plus two more between the endpoints of the interval and the first and last roots. Thus we divide the interval $[a, b]$ in $n+2$ intervals: $[a, z_0]$, $[z_0, z_1]$, ..., $[z_{n-1}, z_n]$, $[z_n, b]$, and locate the points $x_0^*, x_1^*, \ldots, x_{n+1}^*$, where the error function attains its maximum or minimum value. This last step can be done for instance, by computing the root of the derivative of the error function, if such root exists. Otherwise we just compute the error function at the endpoints of the interval.

At this point we have a choice: multi-point exchange, or single-point exchange. Define k such that

$$k = \max_i |f(x_i^*) - p(x_i^*)|.$$

In the single-point exchange we move x_k to x_k^*, while in the multi-point exchange we exchange all $n+2$ points x_i by x_i^*. In the current paper we shall always perform multi-point exchange.

Iteration:

The Remez method then performs steps 1 and 2 until the difference between the old control points and the new ones lies below a given threshold. At the end we have approached the Chebyshev minimax condition, hence the magnitude of the error function at the final set of the $n+2$ control points, E, represents the maximum absolute value of the approximation error.

The steps above are summarized in Algorithm 2.1.

Algorithm 2.1 Remez method

Input: Explicit function $f(x)$, order $n > 0$ and control points $x_0, x_1, \ldots, x_{n+1}$, threshold.

Output: Polynomial $p(x) = c_0 + c_1 x + \cdots + c_n x^n \approx f(x)$ and minimax error E.

1: **while** $|x^{\text{old}} - x^{\text{new}}| >$ threshold **do**

2: Solve the linear system

$$p(x_i) + (-1)^i E = f(x_i) \text{ for } i = 0, \ldots, n+1.$$

3: Compute the roots of the error function $e(x) = f(x) - p(x)$ by the method of chords. Denote them by z_i.

4: Locate the extremum of the error function in every interval $[z_i, z_{i+1}]$. (By computing its derivative, for instance.) Denote them by x_i^{new}.

5: $x_i \leftarrow x_i^{\text{new}}$.

6: **end while**

Numerical Example:

We illustrate the Remez method by finding a fifth order minimax approximation of the function $f(x) = e^x$ on the interval $[-1, 1]$.

We take our initial set of 7 control points uniformly distributed on $[-1, 1]$. Below we show the initial error function, as well as one computed after a certain number of iterations.

2.3.3 The Remez method for rational approximation

If desired, one can extend the Remez algorithm to a rational approximation of the form

$$f(x) \approx R(x) = \frac{p(x)}{q(x)}$$

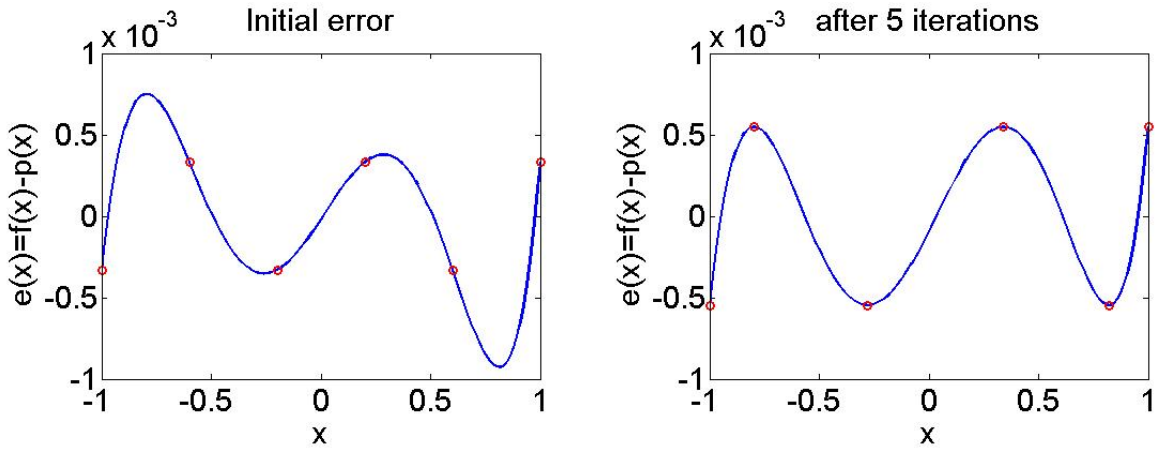

Figure 2.6: Error functions of 5th order minimax approximation with peak value $4.5 \cdot 10^{-5}$.

where $p(x)$ and $q(x)$ are polynomials of degree n and m; see [3]. Assuming $q(x)$ is normalized, i.e., $q(0) = 1$, it will lead to the system with $n + m + 2$ unknowns

$$p(x_i) - f(x_i)q(x_i) + (-1)^i Eq(x_i) = 0, \qquad i = 0, \ldots, n + m + 1,$$

evaluated at $n + m + 2$ control points x_i.

Since the system is nonlinear in the error term, one can adopt an iterative method to solve it. Give an initial guess for E, solve the system for the unknown coefficients, calculate new value of E, and repeat until E converges to a stable value.

Even though the computational effort is considerably increased for rational approximation, it is often desirable to obtain a rational rather than polynomial approximation nonetheless: rational approximations will often match more difficult to approximate functions, to greater accuracy, and with greater efficiency, than their polynomial alternatives. For example, if we take our previous example of an approximation to e^x, we obtain $5.4 \cdot 10^{-4}$ accuracy with an order 4 polynomial. If we move two of the unknowns into the denominator to give a fairly corresponding pair of order 2 polynomials, and re-minimize, then the peak error converges to a value of $1.2 \cdot 10^{-4}$.

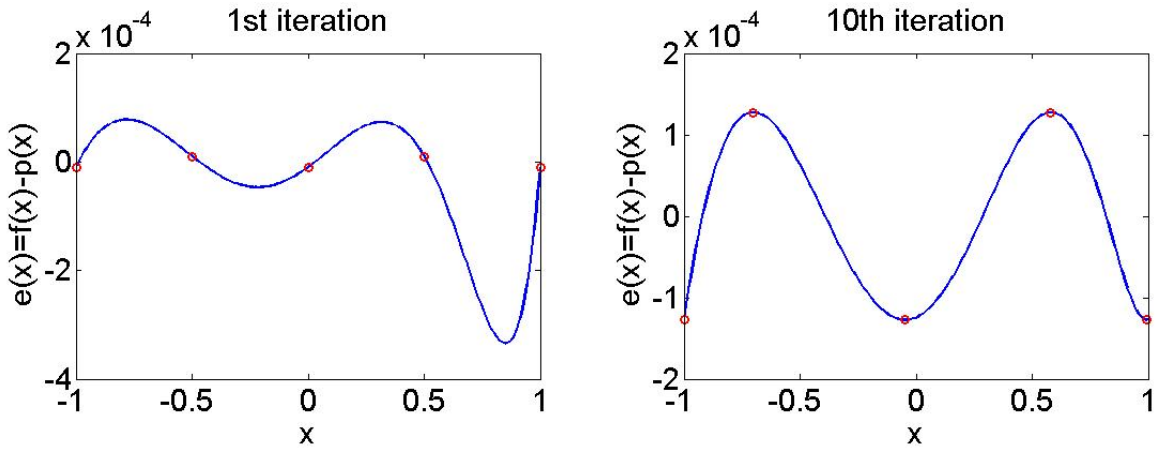

Figure 2.7: Error functions for e^x with given initial minimax error $E = 1e - 5$.

2.4 Overdetermined rational approximation on a complex domain

Up until now, all the previously discussed approximation methods for the exponential e^{-z} were done on the real line. Since one cannot except all eigenvalues of a certain eigenproblem to be real, it is wise to consider a rational type approximation for $f(z) = e^{-z}$ on a complex domain.

Let us consider a box-like domain of approximation Ω, shown in Figure 2.8.

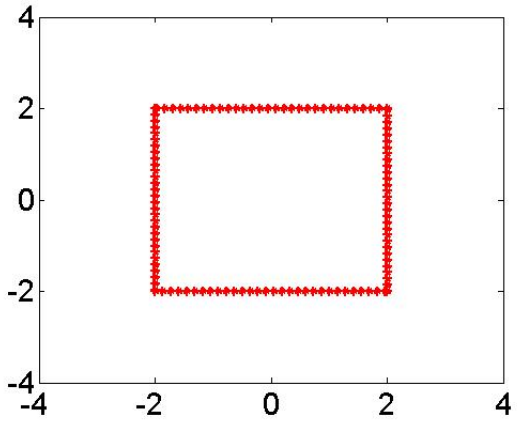

Figure 2.8: Domain for approximation.

Selecting a number of points from Ω, the overdetermined system corresponding to the

rational approximation of e^{-z} will be

$$e^{-z_i} \approx \frac{p_m(z_i)}{q_n(z_i)}, \qquad i = 1, \ldots, N, \tag{2.4}$$

with the additional requirement $N > n + m + 1$, i.e., the number of equations is larger than the number of the unknown coefficients.

A standard technique of finding approximate solutions of such systems is the method of least squares. For this particular case the minimization problem is

$$\min_{\mathrm{C}} \left|\left| \frac{p_m(z_i)}{q_n(z_i)} - e^{-z_i} \right|\right|, \tag{2.5}$$

where C is the set of the polynomial coefficients $\{p_0, p_1, \ldots, p_m, q_1, q_2, \ldots, q_n\}$.

To solve such nonlinear least squares problem the Gauss–Newton algorithm can be applied. Given m functions r_i $(i = 1, \ldots, m)$ of n variables $\mathbf{c} = (c_1, c_2, \ldots, c_n)$, with $m \geq n$, the Gauss-Newton algorithm finds the minimum of the sum of squares

$$S(\mathbf{c}) = \sum_{i=1}^{m} r_i^2(\mathbf{c}).$$

Starting with an initial guess $\mathbf{c}^0$, the method proceeds by the iterations

$$\mathbf{c}^{k+1} = \mathbf{c}^k + \alpha(\delta\mathbf{c}),$$

where α is a damping parameter and the increment $\delta\mathbf{c}$ follows from the normal equation

$$\mathbf{J}(\delta\mathbf{c}) = \mathbf{r},$$

with $\mathbf{r}$ being the vector of functions, in this case residuals r_i, and $\mathbf{J}$ the Jacobian matrix of $\mathbf{r}$ with respect to $\mathbf{c}$, both evaluated at $\mathbf{c}^k$.

For the reader's convenience the Gauss–Newton algorithm is summarized in Algorithm 2.2.

Providing the initial guess from a rational interpolation over $6+6+1$ points and taking 10 iterations of the damped Gauss-Newton algorithm the sum of square of residuals drops to the value $2.0e-9$. The approximations and the corresponding absolute errors are shown in Figure 2.9.

Since the degrees of the approximating (rational) polynomials will have a significant influence on the amount of computational work for the problems discussed in the succeeding

Algorithm 2.2 Gauss–Newton algorithm

Input: vector of functions $\mathbf{r} = (r_1, r_2, \ldots, r_m)$, each having n variables, such that $m \geq n$; initial guess $\mathbf{c}^0$.

Output: solution $\mathbf{c} = (c_1, c_2, \ldots, c_n)$.

1: **for** $k = 0, 1, \ldots$ **do**

2: Compute Jacobian, $\mathbf{J}$, of $\mathbf{r}$ with respect to $\mathbf{c}$.

3: Compute the correction term $\delta\mathbf{c}$ by solving

$$\mathbf{J}(\delta\mathbf{c}) = \mathbf{r}.$$

4: Make the Newton iteration

$$\mathbf{c}^{k+1} = \mathbf{c}^k + \alpha(\delta\mathbf{c}),$$

where the damping parameter $\alpha \in [-1, 10]$ is given such that the $||\mathbf{r}||$ is minimized.

5: **end for**

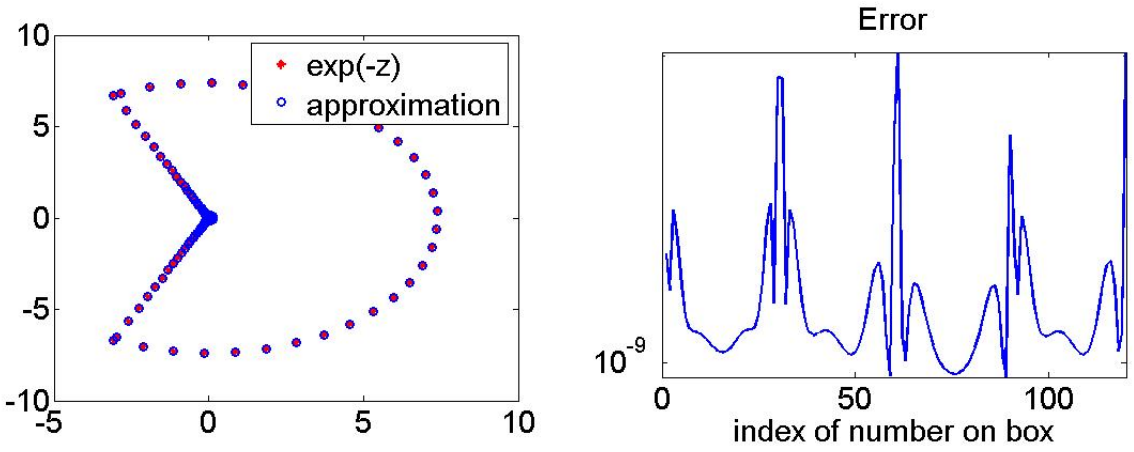

Figure 2.9: Least squares approximation on $\Omega = [-2, 2] + [-2, 2]i$.

chapters, it is desirable to keep them as low as possible. That is, given a tolerance ϵ, we wish to find the degrees m and n of the numerator and the denominator such that the maximum absolute error on a certain complex domain is not larger than ϵ. For reasons that will become clear in the next chapter we consider $m = n + 1$. Thus we consider the problem of finding the degree n such that the given tolerance is reached. Since we expect that this degree depends on the size of the domain of approximations we apply some splittings to the domain Ω and compute the corresponding degree for each of the subdomains. These splittings are made in both the horizontal and the vertical direction simultaneously, such that each subdomain will have the same size. The steps of this process are given in Algorithm 2.3.

Algorithm 2.3 Divide and Conquer

Input: entire domain of approximation $\Omega = [x_1, x_2] + i \cdot [y_1, y_2]$, tolerance ϵ, number of cuts in one direction c, i.e., number of subdomains $(c+1)^2$

Output: degrees $d_k, k = 1, 2, \ldots, (c+1)^2$, such that the given tolerance is reached in each subdomain.

1: Set $x = \text{linspace}(x_1, x_2, c+2)$ and $y = \text{linspace}(y_1, y_2, c+2)$.
2: For each subdomain $\Omega_k, k = 1, 2, \ldots, (c+1)^2$, take initial maximum errors uniformly $e_k = 1$, and starting degrees uniformly $d_k = 2$.
3: **for all** subdomains Ω_k **do**
4: **while** $e_k > \epsilon$ **do**
5: Compute approximation error e_k with Algorithm 2.2.
6: Increase d_k.
7: **end while**
8: **end for**

The result of an application of Algorithm 2.3 for $\Omega = [-10, 20] + i \cdot [-10, 10]$ and $\epsilon = 10^{-2}$ is shown in Figure 2.10.

2.5 Best approximation on the unit disk

Based on certain extensions of the theorem of Carathéodory and Fejér (see [22]), we present in this section a method for finding a best rational approximation to e^{-z} on the unit circle.

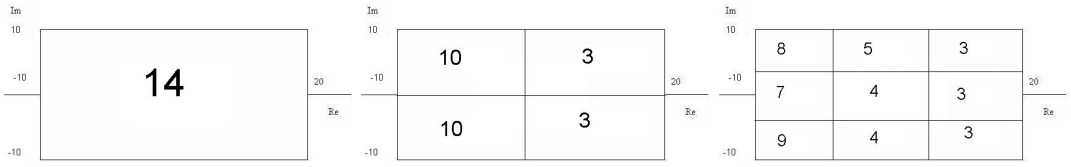

Figure 2.10: Approximation degrees and computation times for the Divide and Conquer approach. Computation times: 35.7, 26.4, and 24.3.

Let $S = \{z \in \mathbb{C} : |z| = 1\}$ be the complex unit circle, let $D = \{z \in \mathbb{C} : |z| < 1\}$ be the open unit disk, and let $\overline{D} = D \cup S$. Let R_{mn} be the space of rational functions of type (m, n), that have no poles in $\overline{D}$. Then, the problem of finding the *best rational approximation* is the following: given f analytic in D and continuous on $\overline{D}$, find a rational function $r^* \in R_{mn}$ such that $\|f - r^*\| = \inf_{r \in R_{mn}} \|f - r\|$ and $|z| \leq 1$.

For given f and any r, the image of S under $f - r$ describes some curve in the plane, which we call the *error curve* corresponding to r. It will be shown that a best approximation r^* is a function whose error curve can be contained in a disk of minimal radius around the origin. Based on the observation in [22], we note that, typically, for smooth f the error curve corresponding to r^* often approximates closely a perfect circle around the origin.

2.5.1 The CF method

We shall now present a method of finding best rational approximations to a given function f. We will name it the *Carathéodory–Fejér approximation* or *CF method*, due to the fact that the original theory of such approximation techniques was first presented by Carathéodory and Fejér, in 1911. For the sake of simplicity, here we shall find approximations $r^* \in R_{nn}$. For the straightforwardly generalized case $m \neq n$, the reader is referred to [22].

Let f be a polynomial, $f(z) = c_0 + c_1 z + \cdots + c_K z^K$, and let H_f denote the *Hankel matrix* of f

$$H_f = \begin{pmatrix} c_1 & c_2 & \dots & c_K \\ c_2 & \dots & c_K & 0 \\ \vdots & \iddots & \iddots & \vdots \\ c_K & 0 & \dots & 0 \end{pmatrix}. \tag{2.6}$$

Note that H_f is symmetric, but if the c_k are not real, it is not Hermitian. Let

$$H_f = U\Sigma V,$$

be the singular value decomposition of H_f, i.e., let the above equation hold with U, V unitary and Σ of the form $\mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\mathrm{K}})$, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_K \geq 0$. Then, the simplified version of the Carathéodory–Fejér theorem from [22] is given in the following proposition; see [22] for a proof.

Proposition 2.2. *The polynomial $f(z) = c_0 + c_1 z + \cdots + c_K z^K$ has a unique best approximation $r^* \in R_{nn}$ on the unit disk. The error is*

$$\|f - r^*\| = \sigma_{n+1}(H_f),$$

where $\sigma_{n+1} = 0$ if $n + 1 > K$, and the error curve is a perfect circle around the origin whose winding number is exactly $2n + 1$. r^* is given by

$$f(z) - r^*(z) = \sigma_{n+1} z^K \frac{u_1 + u_2 z + \cdots + u_K z^{K-1}}{v_K + v_{K-1} z + \cdots + v_1 z^{K-1}}, \tag{2.7}$$

where $u = (u_1, u_2, \ldots, u_K)^T$ is the $(n+1)$st column of U and $v = (v_1, v_2, \ldots, v_K)^T$ is the $(n+1)$st column of V in any singular value decomposition $H_f = U\Sigma V$.

The *winding number* in Proposition 2.2 is an integer representing the total number of times that the error curve travels counterclockwise around the origin. Note, moreover, that Proposition 2.2 is given for functions f of polynomial type. In practice, a given smooth function, such as e^{-z}, may be truncated first to a polynomial by Maclaurin expansion of order K.

Furthermore, we give a final theoretical result of [22]. The following theorem shows that any approximation with a circular error curve must be a best approximation. See [22] for a proof.

Proposition 2.3. *Given f analytic in D and continuous on $\overline{D}$, suppose the error curve of some function $r \in R_{nn}$ is a perfect circle around the origin with winding number $\geq 2n+1$. Then r is a best approximation to f in R_{nn}.*

2.5.2 Numerical computation of r^*

Based on Proposition 2.2, we shall now present a practical algorithm for computing best rational approximants $r^* \in R_{nn}$ to a given function f.

Step 1. First, one must decide at what degree K to truncate the Maclaurin series of f, and then find the $K+1$ required coefficients. The Maclaurin coefficients of a function $f(z)$ can be computed analytically by

$$c_j = \frac{1}{j!} f^{(j)}(0), \qquad \forall j = 0, 1, \ldots, K.$$

Step 2. Find the $(n+1)$st singular value and singular vector of the K by K Hankel matrix H_f.

Step 3. Finally, one must extract the coefficients of r^* from (2.7). (See [22] for details.)

Algorithm 2.4 is given for $f(z) = e^{-z}$.

Algorithm 2.4 Best approximation of e^{-z} on unit circle

Input: order of approximation (n, n).

Output: best rational approximant $r \in R_{nn}$.

1: Discretize $w = \{z \in \mathbb{C} : \ |z| = 1\}$ using at least $2n$ points and set $f(w) = e^{-w}$.

2: Choose degree $K > n$.

3: Compute Maclaurin coefficients

$$c_j = \tfrac{(-1)^j}{j!} \qquad \text{for } j = 0, 1, \ldots, K$$

4: Singular value decomposition of `H = hankel(c)`:

- $[U, S, V] =$ `svd(H)`.
- Select singular value $\sigma_{n+1} = S(n+1, n+1)$.
- Select singular vectors $u = U(:, n+1)$ and $v = V(:, n+1)$.

5: Compute the coefficients of r^* from

$$r^* = f(w) - \sigma_{n+1} w^K \tfrac{u_1 + u_2 w + \cdots + u_K w^{K-1}}{v_K + v_{K-1} w + \cdots + v_1 w^{K-1}}.$$

2.5.3 Numerical example

To illustrate the foregoing results, we shall present how the CF method performs in approximating e^{-z}. Because the magnitude of the Maclaurin coefficients of e^{-z} are decreasing

very rapidly we shall choose for our numerical purposes the rather small truncating value $K = 25$.

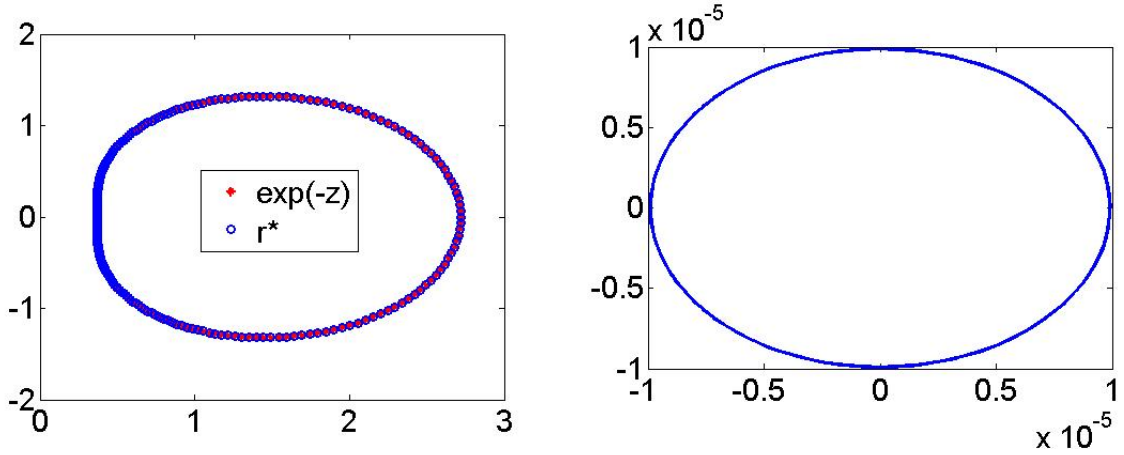

Figure 2.11: (left) Best rational approximation of order $(3, 3)$ and (right) error curve matching a perfect circle with winding number 7, and radius $\sigma_4(H) = 9.93e - 6$.

2.6 Rational Chebyshev approximation in the complex plane

In the previous section we have presented a best approximation method on the unit circle based on Proposition 2.2 by Carathéodory and Fejér. Based on the more recent paper [6] by Ellacott and Williams we wish to present hereby a more general best approximation technique applicable for random complex domains. We are interested in finding a *local best approximation*, defined by the following:

Definition 2.4. *$\frac{p^*}{q^*}$ is a local best approximation to f if for each $p \in P$ and $q \in Q$, there exists a real $T > 0$ such that*

$$\left\| f - \frac{p^*}{q^*} \right\| \leq \left\| f - \frac{p^* + tp}{q^* + tq} \right\|, \qquad \forall t \in [0, T].$$

In the definition above, P and Q refer to the space of complex-valued polynomials of degree at most m and n, respectively. Additionally we define the space of complex rationals

$R := \{p/q : p \in P, q \in Q\}$. Moreover, the term "local" in Definition 2.4 refers to the fact that small perturbations have been made in the numerator and denominator.

Formally, the approximation method presented in [6] is synthesized in Algorithm 2.5.

Algorithm 2.5 Best Chebyshev approximation using Lawson's method

Input: Function f, and discretized boundary $\{z_i\}$ of random complex region.

Output: Best Chebyshev approximation on random complex domain.

1: Choose initial p_0, q_0. (e.g., by Padé approximation.)

2: **for all** $k = 0, 1, \ldots$ **do**

3: Find δp_k, δq_k for which

$$\left\| f - \frac{p_k}{q_k} - \frac{q_k \delta p - p_k \delta q}{q_k^2} \right\| \tag{2.8}$$

is minimized over all polynomials δp and δq. This is done by the so-called *Lawson's method*, presented in Algorithm 2.6.

4: Determine the value t_k from the discretized interval $T = [-1, 1]$, such that

$$\left\| f - \frac{p_k + t_k \delta p_k}{q_k + t_k \delta q_k} \right\|_\infty \leq \left\| f - \frac{p_k + t \delta p_k}{q_k + t \delta q_k} \right\|_\infty, \qquad \forall t \in T.$$

5: Set $p_{k+1} = p_k + t_k \delta p_k$ and $q_{k+1} = q_k + t_k \delta q_k$.

6: **end for**

Before presenting Lawson's method, necessary for computing a best approximation, we give some relevant arguments. First of all, the minimization problem (2.8) is a linearization of the original problem of minimizing

$$\left\| f - \frac{p_k + \delta p}{q_k + \delta q} \right\|.$$

This can be seen by the following steps, in which the $\mathcal{O}(2)$ terms are omitted:

$$\begin{aligned} \frac{p_k + \delta p}{q_k + \delta q} &= \frac{p_k + \delta p}{q_k} \frac{1}{1 + \frac{\delta q}{q_k}} = \frac{p_k + \delta p}{q_k} \left(1 - \frac{\delta q}{q_k} \right) \\ &= \frac{p_k}{q_k} + \frac{q_k \delta p - p_k \delta q}{q_k^2}. \end{aligned}$$

Hence, we need to compute a linear approximation for $e_k = f - \frac{p_k}{q_k}$. As a first step in doing so, we define for a fixed element $p/q \in R$ the set

$$L(p, q) := \left\{ \frac{q\tilde{p} - p\tilde{q}}{q^2} : \tilde{p} \in P, \tilde{q} \in Q \right\}.$$

To emphasize even more on the reasonability of minimizing (2.8) we give a theorem of [6] without proof.

Theorem 2.5. *$p^*/q^* \in R$ is a local best approximation to f if and only if a best approximation to $f - p^*/q^*$ from the space $L(p^*, q^*)$ is the zero function.*

The next theorem gives a basis for $L(p, q)$.

Theorem 2.6. *Let p/q, $q(0) = 1$, be a fixed element in R. Then, $L(p, q)$ is spanned by the $m + n + 1$ elements*

$$\{\varphi_j\} := \left(\frac{zp}{q^2}, \frac{z^2 p}{q^2}, \ldots, \frac{z^n p}{q^2}, \frac{1}{q}, \frac{z}{q}, \ldots, \frac{z^m}{q} \right).$$

Proof: Let $l \in L(p, q)$. Then

$$-l(z) = \frac{p \sum_{j=0}^{n} b_j z^j - q \sum_{j=0}^{m} a_j z^j}{q^2} = b_0 \frac{p}{q^2} + \frac{p \sum_{j=1}^{n} b_j z^j - q \sum_{j=0}^{m} a_j z^j}{q^2}.$$

It is thus sufficient to show that p/q^2 can be written as a linear combination of $\{\varphi_j\}$.

$$\frac{p}{q^2} = \frac{p}{q} + \frac{(1-q)p}{q^2},$$

and so, $\{\varphi_j\}$ spans the space $L(p, q)$. □

In Algorithm 2.6 we give Lawson's method for minimizing (2.8).

Algorithm 2.6 Lawson's method

Input: error $e_k = f - \frac{p_k}{q_k}$.

Output: updates δp_k, δq_k.

1: Set basis functions

$$\Phi_k := \left\{ \frac{zp_k}{q_k^2}, \frac{z^2 p_k}{q_k^2}, \ldots, \frac{z^n p_k}{q_k^2}, \frac{1}{q_k}, \ldots, \frac{z^m}{q_k} \right\}.$$

2: Minimize

$$\left\| e_k - \sum_j \alpha_j \varphi_j \right\|$$

for coefficients α_j and $\varphi_j \in \Phi_k$. (`fminimax`)

3: Set $\delta q_k = [0;\ \alpha(1{:}n)]$ and $\delta p_k = -\alpha(n{+}1{:}n{+}m{+}1)$.

Next we present some numerical results of this method with $f(z) = e^{-z}$ and using different types of complex regions: the unit interval $[-1, 1]$, the unit circle $\{z \in \mathbb{C} : |z| = 1\}$, and the unit square $\{z = x + iy \in \mathbb{C} : x \in [-1, 1], y \in [-1, 1]\}$. The results are summarized in Table 2.1.

$m = n$	$[-1, 1]$	unit disk	unit square
1	$2.1 \cdot 10^{-2}$	$8.4 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$
2	$8.6 \cdot 10^{-5}$	$1.4 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$
3	$1.5 \cdot 10^{-7}$	$1.0 \cdot 10^{-5}$	$3.5 \cdot 10^{-5}$
4	$3.7 \cdot 10^{-12}$	$1.0 \cdot 10^{-7}$	$3.8 \cdot 10^{-7}$

Table 2.1: Approximation errors for e^{-z}, using 100 points for discretization.

Chapter 3

Delay Eigenvalue Problems

3.1 Introduction

Consider the delay-differential equation with a single delay

$$\begin{cases} \dot{x}(t) = A_0 x(t) + A_1 x(t-\tau), & t \geq 0 \\ x(t) = \varphi(t), & t \in [-\tau, 0], \end{cases} \tag{3.1}$$

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.

Just as in the case of ordinary differential equations, the characteristic equation and the corresponding eigenvalues play an important role in the analysis of the delay-differential equation, by making it possible to establish properties of the problem without actually solving it. In the following we give some relevant definitions regarding the characteristic equation of (3.1).

Definition 3.1. *For the DDE (3.1) we call:*

1. *The equation*

$$\det(-\lambda I + A_0 + A_1 e^{-\lambda\tau}) = 0, \tag{3.2}$$

   the characteristic equation *of (3.1);*

2. *a solution λ to the characteristic equation (3.2) an* eigenvalue*;*

3. *the set of all solutions of (3.2) the* spectrum *of the DDE (3.1) and denote it by*

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

4. *a vector* $x \in \mathbb{C}^n, x \neq 0$ *corresponding to the eigenvalue* $\lambda \in \sigma(\Sigma)$*, an* eigenvector *corresponding to* λ *iff*

$$(-\lambda I + A_0 + A_1 e^{-\lambda\tau})x = 0, \tag{3.3}$$

and the pair (λ, x) *is called an* eigenpair*;*

5. *the problem of finding the eigenpairs, i.e., the solution to (3.3), a* delay eigenvalue problem (DEP).

The exponential term in the characteristic equation (3.2) is commonly motivated by looking for non-trivial solutions using the exponential ansatz $x(t) = e^{\lambda t}x_0$. If we insert the exponential ansatz into (3.1) we arrive at the characteristic equation. Because of this nonlinear term, the delay eigenvalue problem (3.3) belongs to a class of problems referred to as *nonlinear eigenvalue problems*.

In this chapter we shall focus on solving the DEP (3.3) by applying some of the approximation methods for the exponential term $e^{-\tau\lambda}$, discussed in Chapter 2.

For our numerical experiments we shall use as a test problem the partial differential equation with delay (PDDE) from [12, Section 2.4.1], i.e.,

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + a_0(x)u + a_1(x)u(x, t-\tau_1) + a_2(x)u(x, t-\tau_2), \\ u(0,t) = u(\pi, t) = 0, t \geq 0 \end{cases} \tag{3.4}$$

where $a_0(x) = a_0 + \alpha_0 \sin(x), a_1(x) = a_1 + \alpha_1 x(1 - e^{x-\pi})$ and $a_2(x) = a_2 + \alpha_2 x(\pi - x)$.

Letting $\tau = \tau_1 = \tau_2$ and selecting the numerical values $a_0 = 20$, $\alpha_0 = 0$, $a_1 = -4$, $\alpha_1 = 1$, $a_2 = -0.1$, $\alpha_2 = 0$ one can discretize (3.4) with center difference and uniform step-size $h = \pi/(n+1)$ in space and get the DDE of dimension n

$$\dot{v}(t) = \frac{(n+1)^2}{\pi^2} \begin{pmatrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & -2 \end{pmatrix} v(t) + \begin{pmatrix} a_0(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_0(x_n) \end{pmatrix} v(t) + \begin{pmatrix} a_1(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_1(x_n) \end{pmatrix} v(t-\tau) + \begin{pmatrix} a_2(x_1) & 0 & \\ 0 & \ddots & 0 \\ & 0 & a_2(x_n) \end{pmatrix} v(t-\tau). \tag{3.5}$$

In the following sections we shall try to approximate the eigenvalues of (3.5) by applying the methods of Chapter 2 for $e^{-\lambda}$, hence $\tau = 1$.

3.2 "Exact" eigenvalues

In order to numerically test our results, i.e., give errors for certain approximate eigenvalues, we need to fix a reference solution of (3.3), which we will consider the "exact" eigenvalues. To this end we introduce the following results of [12].

3.2.1 PDE formulation of a DDE

Every DDE can be rewritten as a partial differential equation by introducing an additional dimension, containing the function segment x_t to the left of the time-point t. This memory-dimension θ takes the position of the space-dimension in the PDE-formulation. We state the equivalence between the DDE and the corresponding PDE by the following theorem; see [12, Section 2.2.3] for a proof.

Theorem 3.2 (PDE-formulation:). *Let $\varphi \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$, that is φ continuous mapping from $[-\tau, 0]$ to $\mathbb{R}^n$, be given. Then consider the single-delay DDE*

$$\begin{cases} \dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) & t \geq 0 \\ x(t) = \varphi(t) & t \in [-\tau, 0] \end{cases} \tag{3.6}$$

and a corresponding boundary value problem

$$\begin{cases} \frac{\partial u}{\partial \theta} = \frac{\partial u}{\partial t} & t \geq 0,\ \theta \in [-\tau, 0] \\ \frac{\partial u}{\partial \theta}(t, 0) = A_1 u(t, -\tau) + A_0 u(t, 0) & t \geq 0 \\ u(0, \theta) = \varphi(\theta) & \theta \in [-\tau, 0] \end{cases} \tag{3.7}$$

with $u \in \mathcal{C}([0, \infty) \times [-\tau, 0], \mathbb{R})$ as the unknown.

Now, suppose $x(t)$ is the solution to (3.6) and $u(t, \theta)$ the solution to (3.7). Then

$$u(t, \theta) = x(t + \theta) \tag{3.8}$$

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

Thus, the solution of the PDE is exactly $x(t)$ shifted over θ. Now let $\mathcal{A}$ correspond to the differentiation operator in θ-direction with the domain of functions fulfilling the boundary conditions in (3.7). That is

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

for functions φ satisfying

$$\varphi'(0) = A_1\varphi(-\tau) + A_0\varphi(0).$$

The boundary value problem (3.7) is hence

$$\frac{d}{dt}x_t = \mathcal{A}x_t, \tag{3.9}$$

with x_t being the solution of the DDE. [12]

3.2.2 Discretizing the PDE

By, for instance, [7, Chapter 7, Lemma 2.1], we have that the eigenvalues of the operator $\mathcal{A}$ are also the eigenvalues of the DDE. The next step in our purpose of finding reference eigenvalues is to discretize $\mathcal{A}$, i.e., the PDE in θ - space - direction and compute the eigenvalues of the corresponding linear operator (matrix) A_N.

Chebyshev spectral differentiation:

There are a number of discretization methods available for such problems. A crude, but rather simple approximation for replacing the differential is by finite differences, e.g. forward differences. Another successful approximation, proven to be highly efficient, is the Chebyshev differentiation matrix.

Consider the grid

$$x_j = \cos\frac{j\pi}{N}, \qquad j = 0, \ldots, N.$$

These points are known as Chebyshev extreme points, since they are the extrema of Chebyshev polynomials

$$T_N(x) = \mathrm{Re}(\mathrm{z^N}) = \frac{1}{2}(\mathrm{z^N} + \mathrm{z^{-N}}).$$

The Chebyshev differentiation matrix, denoted hereby D_N, transforms a vector of data at Chebyshev points into approximate derivatives at those points. Here we will give the

definition of D_N by [23]: let $N \geq 1$ be any integer. The first order spectral differentiation matrix has entries

$$\begin{aligned}
(D_N)_{00} &= \tfrac{2N^2+1}{6}, \\
(D_N)_{NN} &= -\tfrac{2N^2+1}{6}, \\
(D_N)_{jj} &= \tfrac{-x_j}{2(1-x_j^2)}, \qquad 1 \leq j \leq N-1, \\
(D_N)_{ij} &= \tfrac{c_i}{c_j}\tfrac{(-1)^{i+j}}{x_i - x_j}, \quad i \neq j.
\end{aligned}$$

Thus the discretized operator A_N for

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

will be

$$A_N = \begin{pmatrix} & D_N & \otimes & I_n & \\ A_1 & 0 & \dots & 0 & A_0 \end{pmatrix}, \tag{3.10}$$

where $\otimes$ stands for the Kronecker tensor product. Thus, for $(N+1)$ Chebyshev points and coefficient matrices of size $n \times n$, the first block row of A_N corresponding to the differentiation is size $nN \times n(N+1)$, while the last block row, corresponding to the boundary conditions in (3.7) is size $n \times n(N+1)$.

Using the Matlab routine `cheb` from [23], one can elegantly compute the eigenvalues of A_N by

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

where `cN` denotes the dimension of the Chebyshev differentiation matrix and `N` the size of the coefficient matrices A_0 and A_1. Therefore, the first line determines the Chebyshev differentiation matrix for the interval $[-\tau, 0]$, while the second one imposes the boundary conditions in the last row of the differentiation matrix A_N and ultimately computes the eigenvalues of the operator.

Model order reduction

The differentiation operator $\mathcal{A}$ in (3.9) is defined as the infinite-dimensional operator corresponding to the BVP (3.7); see [7] for details. Hence we can say that time-delay systems can be treated as infinite dimensional systems. In this context we wish to note that a rational approximation of $e^{-\tau\lambda}$ can be considered a *model order reduction* from infinite order

to the order of the rational function. This is shown by stating that the discretization of the operator $\mathcal{A}$, presented earlier, could be interpreted as an approximation of the exponential function, at least for some cases.

Let us truncate the limit in the definition of the exponential. That is, replace $e^{-\tau\lambda} = \lim_{N\to\infty}(1+\tau\lambda/N)^{-N}$ with $e^{-\tau\lambda} \approx (1+\tau\lambda/N)^{-N}$ and then multiply (3.3) by $(1+h\lambda)^N$, where $h = \tau/N$. Then we have

$$(A_0(1+h\lambda)^N + A_1)x = \lambda(1+h\lambda)^N x. \tag{3.11}$$

Taking D_N as the forward difference matrix in (3.10), and defining the vectors $x_k = (1+h\lambda)x_{k-1}$ recursively with $x_1 = x$ we get that the constructed vector $(x_1^T, \ldots, x_N^T)$ is clearly an eigenvector of (3.10). The first block row is fulfilled from the recursive construction of x_k and the last row is (3.11).

3.2.3 Reference eigenvalues

At this point we give the reference eigenvalues of our test problem, which shall be used further on for the purpose of estimating approximation errors for several eigenproblem solvers. For our experiments, the dimension of the Chebyshev differentiation matrix is chosen to be `cN = 20`.

To test the approximation one can compute the minimal singular value of

$$-\tilde{\lambda}I + A_0 + e^{-\tau\tilde{\lambda}}A_1$$

for some approximate eigenvalue $\tilde{\lambda}$. Numbers are shown below for rightmost eigenvalues λ_r:

Discr. size	$N = 10$	$N = 40$	$N = 100$
λ_r	19.0068	19.0005	19.0001
$\sigma_{\min}$	$9.20e-11$	$9.16e-11$	$9.16e-11$

Table 3.1: Minimal singular values for rightmost eigenvalue.

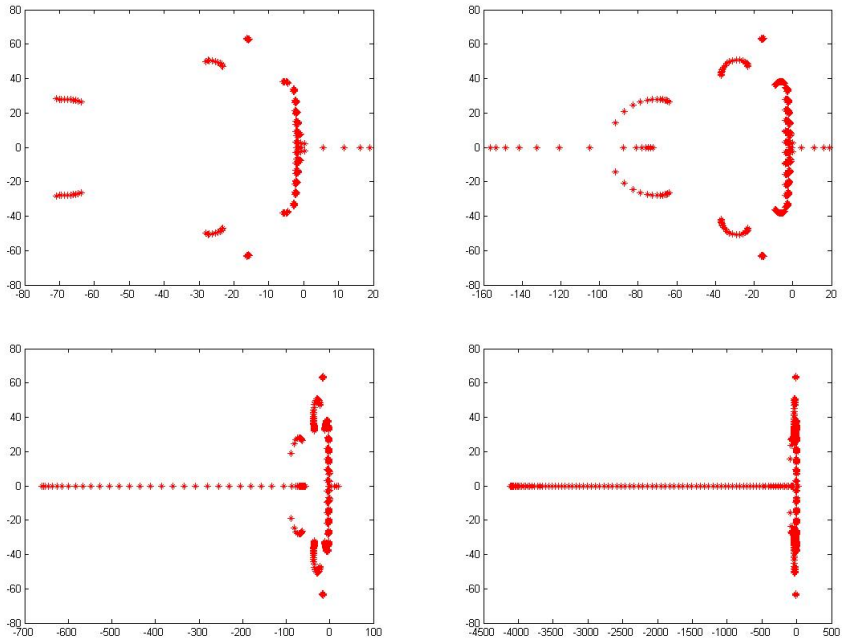

Figure 3.1: Exact eigenvalues for $N = 10$, 20, 40 and 100.

3.2.4 Solution operator discretization

As an alternative method of computing "exact" eigenvalues for the delay eigenproblem one can apply the previously described Chebyshev discretization for the so-called *solution operator*. Below we shall briefly outline the most important findings of [12] with respect to the solution operator without giving any details. For a thorough description the reader is referred to [12, Section 2.2.2].

Definition 3.3. *The* solution operator *of the DDE* (3.1) *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* (3.1) *with initial condition $\varphi = \phi$ at time-point h. More precisely,*

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

where $x(t)$ is the solution of (3.1) *with initial condition $\varphi = \phi$.*

Theorem 3.4. *Consider the DDE* (3.1) *with the solution operator $T(h)$ defined by Defi-*

nition 3.3. Suppose $h \leq \tau$, then for any $\varphi \in \mathcal{C}([-\tau, 0])$,

$$(T(h)\varphi)(\theta) = \begin{cases} \psi(\theta) = \varphi(\theta + h) & \theta \in [-\tau, -h] \\ \textit{Solution of } \dot{\psi}(\theta) = A_0\psi(\theta) + A_1\varphi(\theta + h - \tau) & \theta \in [-h, 0]. \end{cases} \tag{3.12}$$

The operator (3.12) is clearly linear, and the spectrum is related to the spectrum of the DDE by the following *spectral mapping principle*. For any $t > 0$

$$\sigma(\Sigma) = \frac{1}{t} \ln(\sigma(T(t))\backslash 0), \tag{3.13}$$

where $\sigma(\Sigma)$ is the spectrum of the DDE and the logarithm is the set of all branches of the component application on the elements of the set $\sigma(T(t))\backslash 0$.

The following code for the Chebyshev discretization of the solution operator computes some eigenvalues of the single DDE of dimension N with cN Chebyshev nodes.

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

In the last step, the branch k of the logarithm has to be chosen correctly. The rightmost eigenvalues are typically the principal branch $k = 0$; see [12].

Discr. size	$N = 10$	$N = 40$	$N = 100$
λ_r	16.1596	16.0508	16.0437
$\sigma_{\min}$	$5.2e-2$	$4.3e-2$	$4.2e-2$

Table 3.2: Minimal singular values for rightmost eigenvalue.

A comparison of the values from Table 3.1 with the corresponding values from Table 3.2 implies that the method of Section 3.2.2 is more suitable for building the reference eigenvalues.

3.3 Newton-type method

Before turning our attention to several eigenproblems approximating (3.3), we would like to present a Newton-type method for solving eigenvalue problems. Consider the eigenvalue problem

$$G(\lambda)x = 0, \tag{3.14}$$

where G is differentiable. One of the approaches for solving equation (3.14) is to apply Newton's method for the extended system

$$F(u,\theta) = \begin{pmatrix} G(\theta)u \\ w^*u - 1 \end{pmatrix} = 0, \tag{3.15}$$

with given initial eigenpair (u_0, θ_0). The second equation is a normalization condition, where the normalizing vector w, $||w|| = 1$, has to satisfy $w^*x \neq 0$, with x being the exact eigenvector of (3.14). Then the Newton equation

$$0 = F(u_k,\theta_k) + \mathrm{D}F(u_k,\theta_k) \begin{bmatrix} u_{k+1} - u_k \\ \theta_{k+1} - \theta_k \end{bmatrix}$$

for (3.15) at the current approximation (u_k, θ_k) is equivalent to

$$\begin{bmatrix} G(\theta_k) & G'(\theta_k)u_k \\ w^* & 0 \end{bmatrix} \begin{bmatrix} u_{k+1} - u_k \\ \theta_{k+1} - \theta_k \end{bmatrix} = - \begin{bmatrix} G(\theta_k)u_k \\ 0 \end{bmatrix}, \tag{3.16}$$

provided that u_k satisfies the normalizing condition $w^*u_k = 1$.

Hence, given an initial approximate solution to (3.15), one can always refine it by means of the Newton-type method presented hereby. A practical test of the current method is presented in Section 3.5.2. For further details on Newton-type methods for nonlinear eigenvalue problems the reader is referred to [17].

3.4 Galerkin and minimal residual method

Assume we have an approximate eigenvector u of the eigenproblem (3.3) and we want to determine the corresponding approximate eigenvalue θ. In this section we present suitable methods for doing so by extending the results of [11, Chapter 8] on quadratic eigenvalue problems to delay eigenvalue problems of type (3.3).

3.4.1 Galerkin method

For an approximate eigenpair (θ, u) of (3.3) we define the *residual* $r(\theta, u)$ by

$$r(\theta, u) := (-\theta I + A_0 + e^{-\tau\theta} A_1)u.$$

Then the approximate eigenvalue θ follows from imposing the *Ritz–Galerkin* condition $r(\theta, u) \perp u$, i.e.,

$$-\theta + u^* A_0 u + e^{-\tau\theta} u^* A_1 u = 0. \tag{3.17}$$

One standard approach for solving this problem directly is based on the *Lambert W* function. The Lambert W function, denoted $W_k(z)$, is a logarithmic type function defined as the multivalued inverse of the complex function $f(w) = we^w$:

$$W_k(z) \in \{w \in \mathbb{C} : z = we^w\}, \qquad k \in \mathbb{Z}, \tag{3.18}$$

where k denotes one of the infinite number of branches. The solution to (3.17), based on the Lambert W function, is then given by the following steps. Introducing $a_0 := u^* A_0 u$ and $a_1 := u^* A_1 u$, (3.17) rewrites to

$$-\theta + a_0 + e^{-\tau\theta} a_1 = 0. \tag{3.19}$$

Multiplying (3.19) by $\tau e^{\tau\theta}$, and denoting $\theta - a_0 =: \psi$ one obtains

$$\begin{aligned} \tau(\theta - a_0)e^{\tau\theta} &= \tau a_1, \\ \tau\psi e^{\tau\psi} &= \tau a_1 e^{-\tau a_0}, \\ \tau\psi &= W_k(\tau a_1 e^{-\tau a_0}). \end{aligned}$$

Hence, the solution θ of (3.17) is

$$\theta = \frac{1}{\tau} W_k(\tau a_1 e^{-\tau a_0}) + a_0. \tag{3.20}$$

In this section we introduce some new methods for solving (3.17). To this order let us define the generalized residual $r(\theta, \eta, u)$ (cf. [11, Section 8.2.2]) by

$$r(\theta, \eta, u) := (-\theta I + A_0 + \eta A_1)u. \tag{3.21}$$

Now one can impose two Galerkin conditions $r(\theta,\eta,u)\perp w_1$ and $r(\theta,\eta,u)\perp w_2$ for specific independent vectors w_1, w_2, leading to the system

$$W^*Z\begin{bmatrix}\theta\\ \eta\end{bmatrix} = -W^*A_0u, \qquad \text{where} \qquad W=[\,w_1 \;\; w_2\,], \qquad Z=[\,-u \;\; A_1u\,]. \tag{3.22}$$

When W^*Z is nonsingular, (3.22) defines unique θ and η approximating λ and $e^{-\tau\lambda}$, respectively. A logical choice for w_1 and w_2 is any linear combination of $-u$, A_0u, and A_1u. Specifically, for the two-dimensional Galerkin method, one can take the two "largest" left singular vectors of $[\,-u \;\; A_1u \;\; A_0u\,]$; see [11] for details.

3.4.2 Minimum residual method

Another idea is to minimize the norm of the generalized residual (3.21) with respect to θ,η:

$$\underset{(\theta,\eta)\in\mathbb{C}^2}{\operatorname{argmin}} ||(-\theta I + A_0 + \eta A_1)u||.$$

To solve this consider the corresponding overdetermined $n\times 2$ linear system

$$Z\begin{bmatrix}\theta\\ \eta\end{bmatrix} = -A_0u,$$

with $Z=[\,-u\; A_1u\,]$. $-u$ and A_1u being independent, θ and η are uniquely determined by

$$\begin{bmatrix}\theta\\ \eta\end{bmatrix} = -(Z^*Z)^{-1}Z^*A_0u. \tag{3.23}$$

One can notice that (3.23) is a special case of (3.22), namely the case when we choose $W=Z$.

Hence, given an approximate eigenvector u, the two-dimensional Galerkin and minimum residual methods provide an approximate eigenvalue λ. We note that one can determine λ in two different ways. Besides the obvious choice $\lambda=\theta$, one can also take

$$\lambda = \underset{\lambda}{\operatorname{argmin}} ||(\lambda, e^{-\tau\lambda}) - (\theta,\eta)||^2,$$

which can be easily computed by Matlab's `fminsearch`. Note, moreover, that for small systems one can always select the better approximation by computing the minimal singular value of (3.3) for the given eigenvalue.

Algorithm 3.1 Two-dimensional methods

Input: an approximate eigenvector u.

Output: two approximate eigenvalues of (3.3).

1: Choose plane of projection spanned by w_1 and w_2:

- **Galerkin:** the two "largest" left singular vectors of $[\ -u\ A_1u\ A_0u\ ]$,
- **Minimum residual:** $w_1 = -u$ and $w_2 = A_1u$.

2: Compute $(\theta, \eta) = -(W^*Z)^{-1}W^*A_0u$, where $W = [\ w_1\ w_2\ ]$ and $Z = [\ -u\ A_1u\ ]$.

3: Approximate λ by

- θ or
- $\operatorname{argmin}_\lambda ||(\lambda, e^{-\tau\lambda}) - (\theta, \eta)||^2$

For the readers convenience we summarize the two-dimensional methods in Algorithm 3.1. For numerical experiments the reader is referred to Section 3.5.2.

Additionally, one can always use the techniques described beforehand for refining an approximate eigenpair (θ, u) of (3.3). This is done in two easy steps: first, compute a new eigenvalue $\tilde{\theta}$ using one of the methods described in Section 3.4 for u, and second, compute the "smallest" singular vector $\tilde{u}$ of $-\tilde{\theta}I + A_0 + e^{-\tau\tilde{\theta}}A_1$ for the approximate eigenvalue $\tilde{\theta}$. Repeating these two steps leads to Algorithm 3.2, at the end of which it is expected to have a better approximation of the eigenpair. For numerical results see Section 3.5.2.

Algorithm 3.2 Refinement of an approximate eigenpair

Output: refined approximate eigenpair.

1: Compute approximate eigenpair from any approximate eigenproblem
2: **for** $k = 1, 2, \ldots$ **do**
3: Compute approximate θ_k by Galerkin or minimum residual method
4: Compute the "smallest" singular vector u_k of $-\theta_k I + A_0 + e^{-\tau\theta_k}A_1$
5: **end for**

3.5 Approximate eigenproblems

In this section we shall apply some of the approximations derived in Chapter 2 to the exponential term in (3.3), and subsequently solve the resulting eigenvalue problem. This eigenvalue problem can be of different types with respect to the approximation on hand. Hence, a polynomial approximation of the exponential will lead to a *polynomial eigenvalue problem*, while a rational approximation will result in a *rational eigenvalue problem*, respectively. The polynomial eigenvalue problem, hereinafter PEP, can be solved using standard techniques, e.g. companion linearization. Furthermore, the rational eigenvalue problems can be trivially rewritten to a polynomial eigenvalue problem by multiplying both sides of (3.3) with the denominator. Thus the method of companion linearization for solving polynomial eigenvalue problems will be extensively used in this thesis (mainly by the `polyeig` routine of Matlab), so it is advisable to briefly present it in the following section.

3.5.1 Linearization of polynomial eigenproblems [12]

The problem of determining $\lambda \in \mathbb{C}$ and $v \in \mathbb{C},\ v \neq 0$ such that

$$P(\lambda)v = (A_0 + A_1\lambda + \cdots + A_N\lambda^N)v = 0,$$

where $A_0, A_1, \ldots, A_N \in \mathbb{C}^{n\times n}$, is called a polynomial eigenvalue problem. A common way to solve such problems is to transform it to a generalized eigenvalue problem. This is done by considering a linearization for $P(\lambda)$ of the form $L(\lambda) = \lambda X + Y$, such that there are some unimodular $E(\lambda),\ F(\lambda)$ with

$$E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} P(\lambda) & 0 \\ 0 & I \end{pmatrix}.$$

In this context a unimodular matrix is defined as a square integer matrix with determinant ± 1. The most common linearization (used by `polyeig` as well) is the so-called first companion linearization $L(\lambda) = \lambda X + Y$, where the matrices $X, Y \in \mathbb{C}^{Nn\times Nn}$ are given by

$$X = \begin{pmatrix} A_N & & & \\ & I & & \\ & & \ddots & \\ & & & I \end{pmatrix}, \qquad Y = \begin{pmatrix} A_{N-1} & A_{N-2} & \ldots & A_0 \\ -I & 0 & \ldots & 0 \\ & \ddots & \ddots & \vdots \\ & & -I & 0 \end{pmatrix}.$$

Thus the solution process for PEP is done by first linearizing $P(\lambda)$ into $L(\lambda)$, and then solving the generalized eigenproblem $L(\lambda)v = 0$, which straightforwardly gives us the eigenvalues of $P(\lambda)$.

3.5.2 REP by Padé approximations for $e^{-\tau\lambda}$

In this section we shall apply some Padé approximations to the exponential term in (3.3) and solve the corresponding rational eigenvalue problem, or REP.

$(2,2)^n$ Padé approximation

Consider the $(2,2)^n$ Padé approximation given by

$$e^{-\tau\lambda} \approx r_n(\lambda) = \frac{p(\lambda)}{q(\lambda)} \tag{3.24}$$

where $p(\lambda) = \left(1 - \frac{\tau\lambda}{2n} + \frac{\tau^2\lambda^2}{12n^2}\right)^n$ and $q(\lambda) = \left(1 + \frac{\tau\lambda}{2n} + \frac{\tau^2\lambda^2}{12n^2}\right)^n$, such that the approximation error is bounded by

$$|e^{-\lambda} - r_n(\lambda)| \leq \frac{1}{270}\left|\frac{\lambda}{n}\right|^5 n3^{-\mathrm{Re}\lambda} \tag{3.25}$$

for $|\lambda/n| \leq 1$ and $n > -\mathrm{Re}(\lambda)$; see [16].

Applying (3.24) to approximate the exponential function in (3.3) we arrive to the REP

$$(-\lambda I + A_0 + r_n(\lambda)A_1)x = 0$$

which, written as a PEP, can be solved by companion linearization. The PEP follows from multiplying the REP with $q(\lambda)$:

$$(-\lambda q(\lambda)I + q(\lambda)A_0 + p(\lambda)A_1)x = 0. \tag{3.26}$$

We solve (3.26) using MatLab routine `polyeig` for matrix coefficients computed by

$$B(2n+1-i) = p_iA_1 + q_iA_0 + \tilde{q}_iI, \qquad i = 2n+1, 2n, \ldots, 1, 0, \tag{3.27}$$

where p_i, q_i and $\tilde{q}_i$ are the coefficients of the polynomials $p(\lambda)$, $q(\lambda)$ and $\lambda q(\lambda)$, respectively.

Results are shown in Figure 3.2, where the convergence order q is calculated by

$$q = \frac{\log \frac{|\lambda_k - \lambda_{k+1}|}{|\lambda_{k+1} - \lambda_{k+2}|}}{\log \frac{|\lambda_{k+1} - \lambda_{k+2}|}{|\lambda_{k+2} - \lambda_{k+3}|}},$$

where the λ_k are approximations of the rightmost eigenvalue of (3.3). The formula above follows from the definition of the rate of convergence

$$\mu = \lim_{k \to \infty} \frac{|\lambda_k - \lambda_{k+1}|}{|\lambda_{k+1} - \lambda_{k+2}|^q}.$$

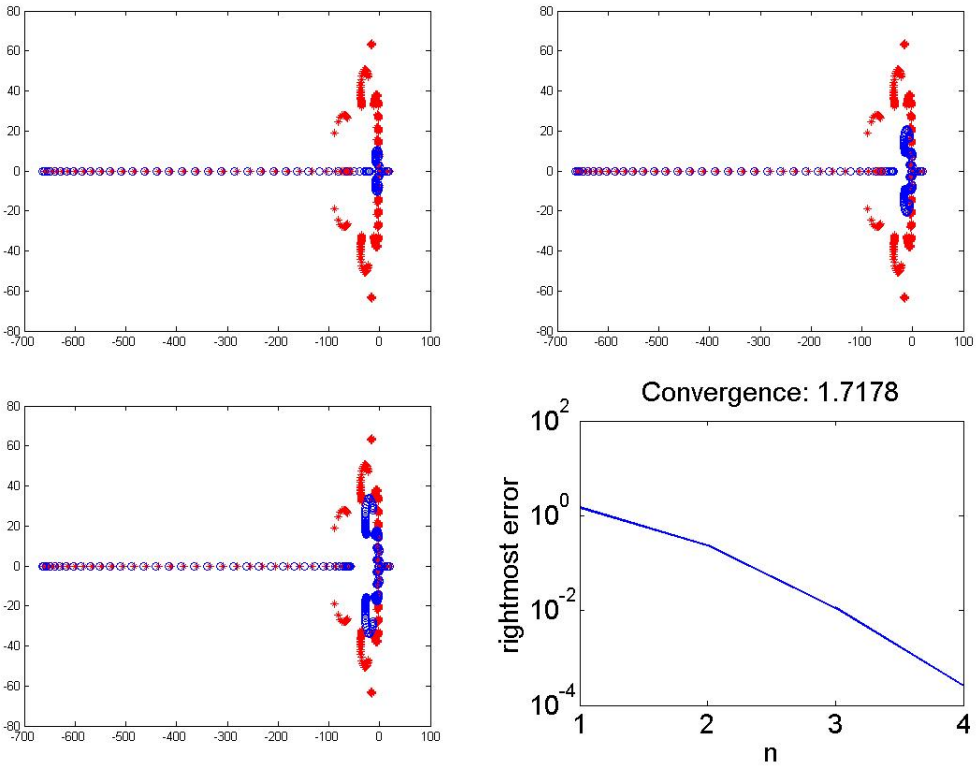

Figure 3.2: Approximate eigenvalues for $n = 2, 3$ and 4 and convergence order for the rightmost eigenvalue $\lambda_r = 19.0005$. Dimension $N = 40$.

The approximation errors for the rightmost eigenvalue of the $N = 20$ problem are given in Table 3.3.

$(2n, 2n)$-Padé vs. $(2, 2)^n$-Padé

Since we have compared the two types of Padé approximation for e^{-z} in the previous chapter, it is also favorable to compare the eigenvalues resulting from the two corresponding

Approximation	$(2,2)$	$(2,2)^2$	$(2,2)^3$	$(2,2)^4$
Error	1.5	$2.3 \cdot 10^{-1}$	$1.1 \cdot 10^{-2}$	$3.0 \cdot 10^{-4}$

Table 3.3: Approximation errors for $N = 20$.

rational eigenvalue problems.

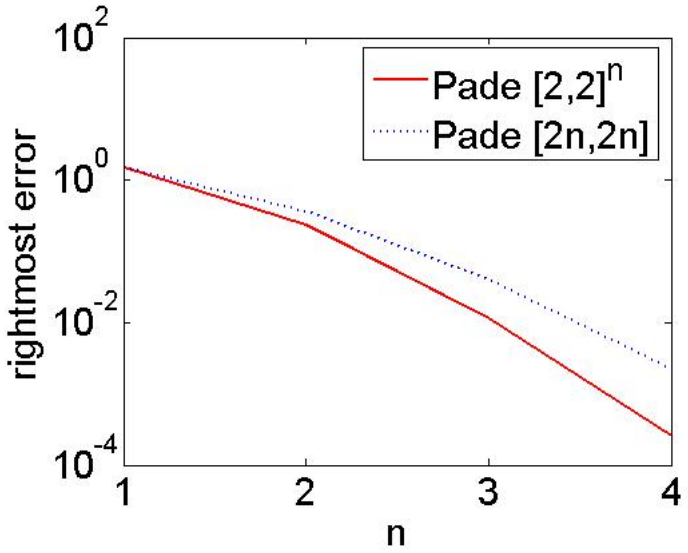

Figure 3.3: Errors for different Padé approximations

As seen in Figure 3.3, the absolute errors for the rightmost eigenvalue are smaller in the case of the $(2,2)^n$-Padé approximation. This result perfectly fits the findings of Section 2.1.2, since the righmost eigenvalue, in the current case $\lambda_r = 19.0005$, is considerably far from the origin.

Newton refinement for rightmost eigenvalue

We would like to show in this section how powerful the Newton-type method, presented in Section 3.3, really is. Consider the nonlinear eigenvalue problem

$$G(\lambda)\mathbf{x} = 0, \tag{3.28}$$

being our original delay eigenvalue problem with

$$G(\lambda) = -\lambda I + A_0 + e^{-\tau\lambda}A_1.$$

Taking the rightmost eigenpair of the polynomial eigenproblem (3.26) as the initial approximation for the eigenpair of (3.28) one can apply Newton's method for refining it.

Considering a $(2,2)^4$ Padé-type approximation for the exponential and applying the Newton iteration 10 times, the minimal singular value for the rightmost eigenvalue drops from $2.55e-4$ to $6.75e-14$. Notice that this value is even smaller than the minimal singular value for our so-called "exact" rightmost eigenvalue. (See Table 3.1 for $N = 40$.)

Numerical results for the Galerkin and minimal residual method

Hereby we wish to present some results regarding the utility of the Galerkin and minimal residual methods presented in Section 3.4. First we test the methods for two different approximate eigenvectors. Initially, we consider as approximate eigenvector the rightmost eigenvector u of (3.26) with polynomials p and q, both of degree 2^4. Furthermore, we will consider another approximate eigenvector u^* by adding a random perturbation to u. Thus

$$u^* = \frac{u + \epsilon w}{||u + \epsilon w||},$$

where w is a normalized vector of the form `rand(N,1)-0.5` $+ i \cdot$ (`rand(N,1)-0.5`), and ϵ will be chosen freely.

Results are summarized in Table 3.4. Note that the error in Table 3.4 refers to the absolute difference from the original rightmost eigenvalue of (3.26), $\lambda_r = 19.0002$. Since the two solutions from Section 3.4, i.e., θ and $\text{argmin}_\lambda ||(\lambda, e^{-\tau\lambda}) - (\theta, \eta)||^2$ will be exactly the same for the case on hand, we shall only give the corresponding errors and minimal singular values one time. Moreover, in some cases, the eigenvalue given by the Lambert W function is more accurate than the eigenvalue resulting from solving (3.26) by `polyeig`. These values are marked with bold.

	Galerkin		Minimum residual		Lambert W	
ϵ	error	$\sigma_{\min}$	error	$\sigma_{\min}$	error	$\sigma_{\min}$
0	$8.9e-7$	$2.5e-4$	$8.9e-7$	$2.5e-4$	**2.5e − 4**	**1.1e − 10**
10^{-6}	$9.5e-6$	$2.6e-4$	$3.4e-6$	$2.5e-4$	**2.5e − 4**	**4.8e − 10**
10^{-3}	$9.4e-1$	$9.4e-1$	$7.8-3$	$7.8e-3$	$1.1e-4$	$3.7e-4$

Table 3.4: Errors and minimal singular values.

One can conclude that the two-dimensional Galerkin and the two-dimensional minimum residual methods provide similarly good approximations for the eigenvalue of (3.3). For increased perturbations (higher ϵ) the minimum residual method gives slightly better results.

Next we wish to show how the refinement technique, presented in Section 3.4, works for an approximate eigenpair (θ, u), being the rightmost eigenpair of (3.26) for p and q of order 2^4. We perturb u using $\epsilon = 10^{-3}$, and follow the steps of Algorithm 3.2 for the Galerkin method. Results are shown in Table 3.5.

iteration	1	2	3	4
error	$3.7e-1$	$2.5e-4$	$2.5e-4$	$2.5e-4$
$\sigma_{\min}$	$3.7e-1$	$5.2e-9$	$9.2e-13$	$2.0e-15$

Table 3.5: Refinement for an approximate eigenvalue.

The poor results of the first iteration are due to the fact that initially we have taken our approximate eigenvector as $u = u_{\text{pep}} + \epsilon w$, where u_{pep} is already nothing else than an approximation of the eigenvector resulting from the polynomial eigenvalue problem (3.26). The next iteration, however, already provides a stable value for the error $|\lambda_{\text{pep}} - \theta|$.

3.5.3 REP by Padé approximations for $\ln \mu$

Consider the change of variables

$$\mu = e^{-\tau\lambda}. \tag{3.29}$$

Thus (3.3) will have the form

$$\left(\frac{1}{\tau}\ln\mu I + A_0 + \mu A_1\right) v = 0. \tag{3.30}$$

This eigenproblem, containing a nonlinear logarithmic term, can again be approximated by a REP, using the techniques of rational or polynomial approximation for $\ln\mu$. One such technique considered here is a $(2,2)$-Padé approximation for $\ln\mu$

$$\ln\mu \approx \frac{\mu^2 - 1}{\frac{1}{3}(\mu^2 + 4\mu + 1)}. \tag{3.31}$$

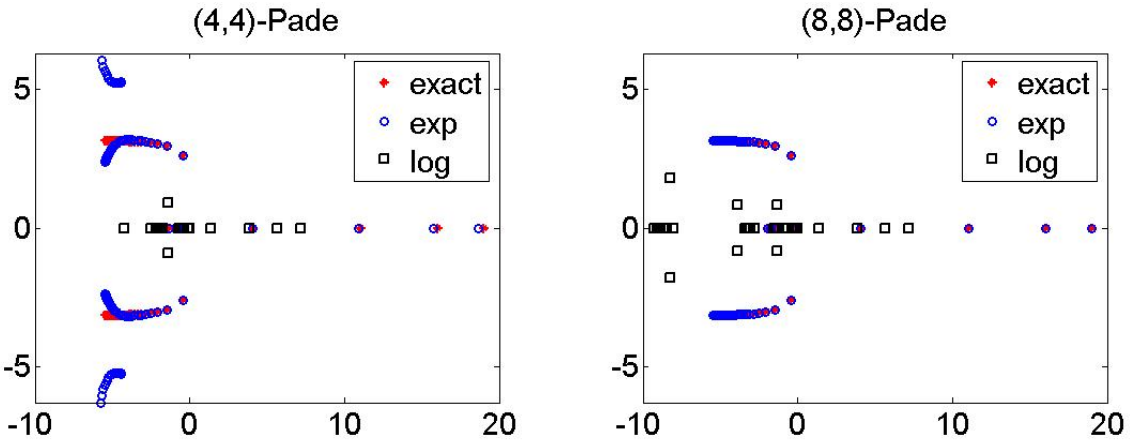

Figure 3.4: Eigenvalues for different Padé approximations of $e^{-\lambda}$ and $\ln \mu$.

Substituting (3.31) in (3.30) we get a polynomial eigenproblem, which can be solved again by `polyeig`.

It can be concluded by Figure 3.4, that the rational approximation of the eigenproblem (3.30) yields worse results than the corresponding approximation of (3.3).

3.5.4 REP by Chebyshev and Legendre interpolation

Considering a rational interpolation of $e^{-\tau\lambda}$ on $[-1, 1]$ as described in the previous chapter one can compute approximate eigenvalues of (3.3). Figure 3.5 clearly shows that these are not at all satisfactory.

3.5.5 PEP by minimax approximation

Consider again the DEP

$$(-\lambda I + A_0 + e^{-\tau\lambda} A_1)x = 0.$$

This eigenvalue problem can be directly transformed to a PEP by applying a polynomial approximation for the term $e^{-\tau\lambda}$. One such method of approximation discussed in the previous chapter is the so-called Remez algorithm.

As shown in Figure 3.6, a 10th order polynomial approximation of the exponential having minimax error $1.46 \cdot 10^{-4}$, yields fairly disappointing results.

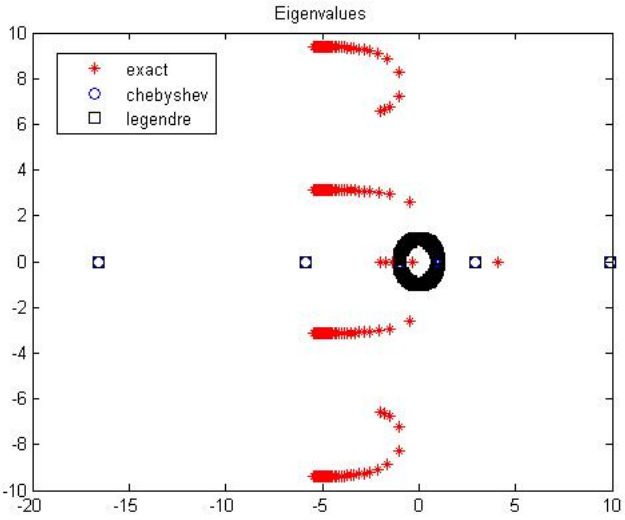

Figure 3.5: Approximate eigenvalues resulting from rational interpolations of order $(30, 30)$ with maximum residuals of $7.6e-7$ and $3.9e-7$, respectively.

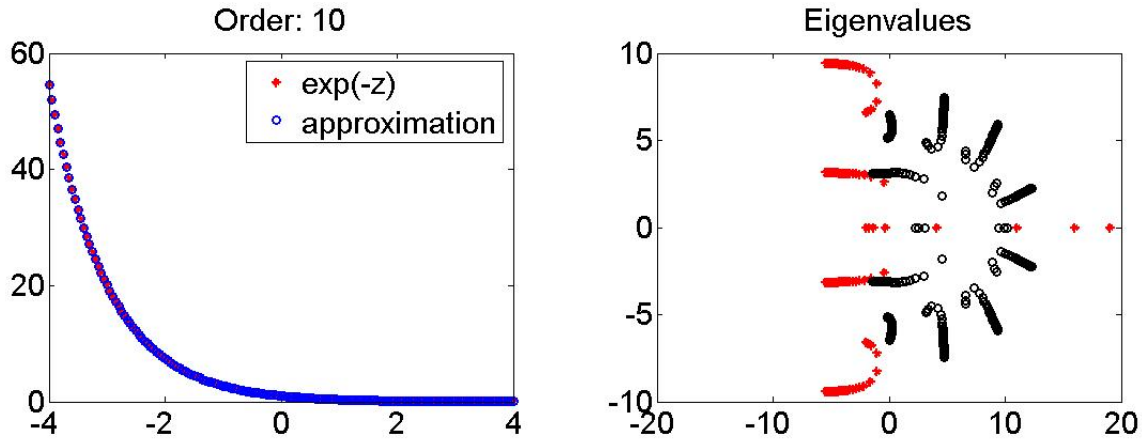

Figure 3.6: Remez approximation to e^{-z} and the corresponding eigenvalues.

3.5.6 PEP by Lagrange interpolation

Applying Lagrange interpolation for $e^{-\tau\lambda}$ yields, yet again, a polynomial eigenproblem. One can observe on Figure 3.7 that the corresponding eigenvalues can be considered fair approximations of the eigenvalues around the origin. However, the rightmost eigenvalue of (3.3) is still very far from being approximated.

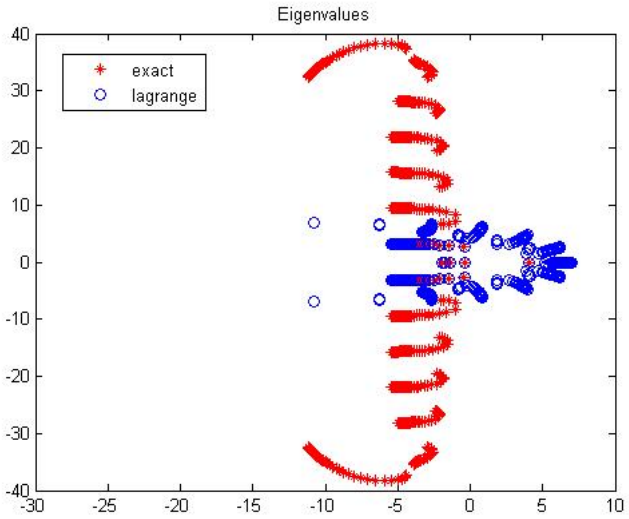

Figure 3.7: Eigenvalues following Lagrange interpolation of order 16 on $[-10, 5]$.

3.5.7 REP by rational minimax approximation

Using the findings on the Remez method for rational approximation, presented in Section 2.3.3, one can rewrite (3.3) as a REP, and consequently as a PEP, which is solved by means of `polyeig`. Figure 3.8 shows approximate eigenvalues given by a $(4, 4)$ rational Remez approximation of e^{-z} on $[-1, 1]$. For the first time the approximation error of the rightmost eigenvalue of (3.3) is one worth mentioning: $7.2 \cdot 10^{-1}$.

3.5.8 REP by least squares approximation on complex box

Applying the approximation method presented in Section 2.4 (on $\Omega = [-2, 2] + i \cdot [-2, 2]$) one finds relatively sharp approximate eigenvalues of (3.3); see Figure 3.9. The errors of approximation for the rightmost eigenvalue of (3.3) are given in Table 3.6.

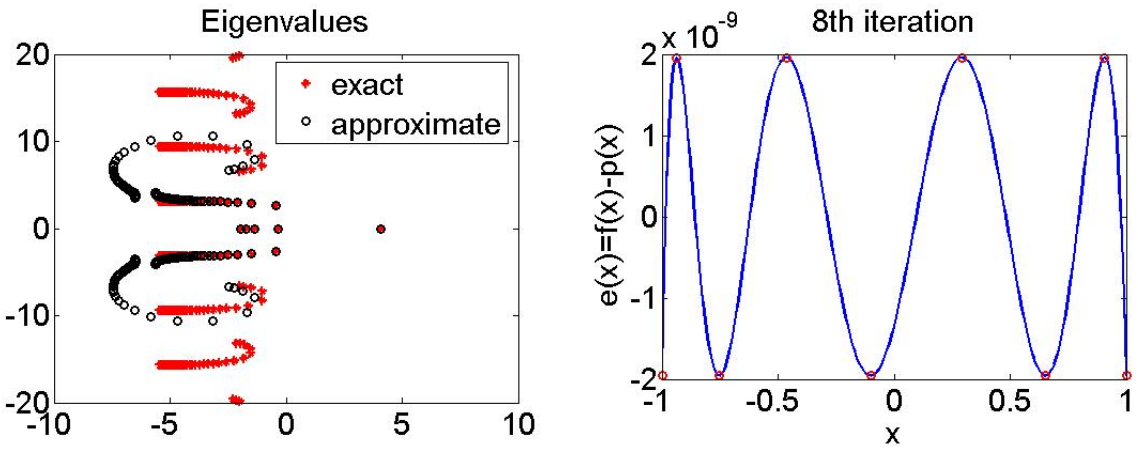

Figure 3.8: Eigenvalues by rational Remez approximation. Problem size $N = 40$.

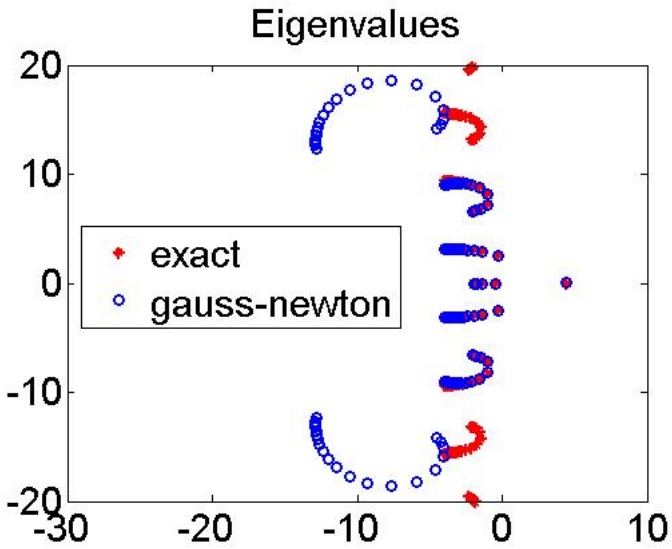

Figure 3.9: Approximate eigenvalues for $N = 20$ and approximation order $(6, 6)$.

order of approximation	$(2,2)$	$(4,4)$	$(6,6)$	$(8,8)$
error	3.7	$6.7 \cdot 10^{-1}$	$6.7 \cdot 10^{-2}$	$7.0 \cdot 10^{-3}$

Table 3.6: Approximation errors for rightmost eigenvalue.

3.5.9 REP by best rational approximation on unit disk

In this section we solve the delay eigenvalue problem (3.3) with the help of the Carathédory–Fejér method described in Section 2.5. Figure 3.10 shows that using a best rational approximation for $e^{-\tau\lambda}$ on the unit circle one finds reasonably accurate approximate eigenvalues for (3.3). The rightmost approximation errors for this method are given in Table 3.7.

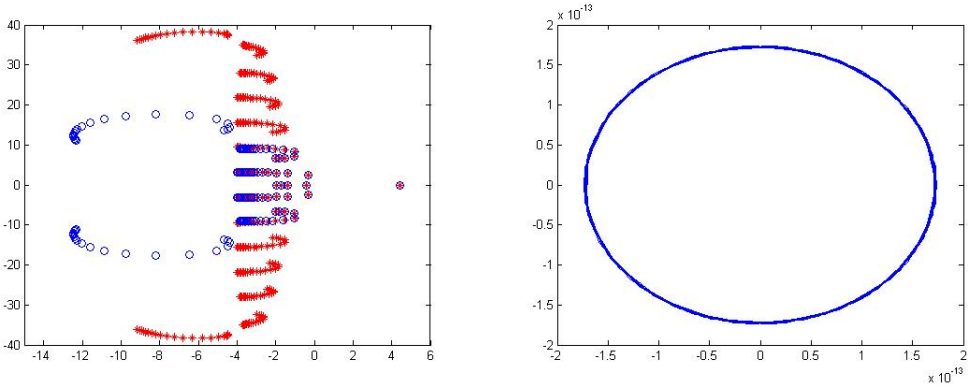

Figure 3.10: (left) Approximate eigenvalues for problem dimension $N = 20$ and approximation order $(6, 6)$. (right) Error curve for the rational approximation of $e^{-\tau\lambda}$ having radius $1.7 \cdot 10^{-13}$.

order of approximation	$(2, 2)$	$(4, 4)$	$(6, 6)$	$(8, 8)$
error	1.8	$4.0 \cdot 10^{-1}$	$4.4 \cdot 10^{-2}$	$2.5 \cdot 10^{-3}$

Table 3.7: Approximation errors for rightmost eigenvalue.

3.5.10 REP by best rational approximation on random complex regions

Figure 3.11 shows results of solving the rational eigenvalue problem resulting from a rational Chebyshev approximation for $e^{-\tau\lambda}$ on several complex domains. Table 3.8 gives the corresponding errors of the rightmost approximate eigenvalue.

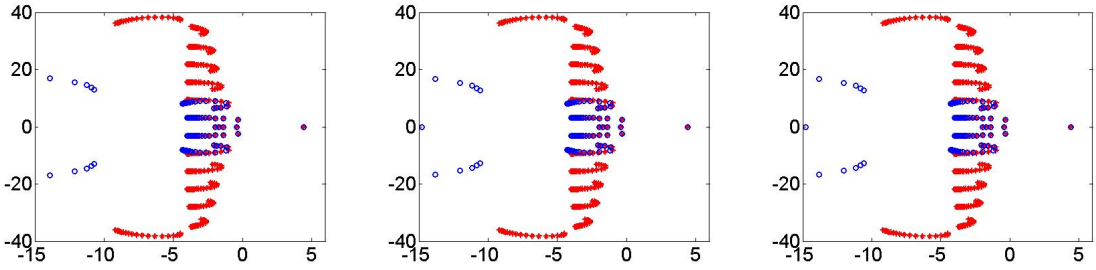

Figure 3.11: Approximate eigenvalues for problem dimension $N = 20$ and approximation order $(5, 5)$ by rational Chebyshev approximation for (1) the interval $[-1, 1]$, (2) the unit disk, and (3) the unit square.

order of approximation	$(2, 2)$	$(4, 4)$	$(6, 6)$	$(8, 8)$
$[-1, 1]$	1.6	$3.7 \cdot 10^{-1}$	$4.0 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$
unit disk	1.8	$3.5 \cdot 10^{-1}$	$4.0 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$
unit square	1.8	$4.0 \cdot 10^{-1}$	$4.6 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$

Table 3.8: Approximation errors for rightmost eigenvalue.

Chapter 4

Perturbation theory

4.1 Introduction

In this chapter we are concerned with defining important tools for the perturbation analysis of the delay eigenvalue problem, such as the normwise backward error, condition numbers and pseudospectrum. Consider the nonlinear delay eigenvalue problem (DEP)

$$(-\lambda I + A_0 + e^{-\tau\lambda}A_1)x = 0, \tag{4.1}$$

where $I, A_0, A_1 \in \mathbb{C}^{N\times N}$, and τ is the delay parameter.

The importance of backward errors for investigating the stability and quality of numerical algorithms and condition numbers for characterizing the sensitivity of solutions to problems is widely appreciated. The forward error, condition number and backward error are related by the inequality

$$\textit{forward error} \ \leq \ \textit{condition number} \ \times \ \textit{backward error}.$$

The theory of backward error and conditioning is well developed for the generalized eigenvalue problem (see [8]), the polynomial eigenvalue problem (see [19]) and the multiparameter eigenvalue problem as well (see [11, Chapter 7]). The definitions and theorems of the current chapter are simply reformulations of the ones appearing in the aforementioned works.

4.2 Preliminaries

Throughout this chapter we consider the arbitrary matrices E_l for $l = 0, 1$, representing tolerances against which the perturbations ΔA_0 and ΔA_1 of A_0 and A_1 are measured. More explicitly, we have for a given $\epsilon > 0$, that $||\Delta A_0|| \leq \epsilon ||E_0||$ and $||\Delta A_1|| \leq \epsilon ||E_1||$. Usually we take either $E_l = A_l$, considering normwise relative perturbations, or $E_l = I$, considering normwise absolute perturbations.

For further convenience we introduce the notation

$$D(\lambda) := -\lambda I + A_0 + e^{-\tau\lambda} A_1,$$

and define

$$\Delta D(\lambda) := -\lambda I + \Delta A_0 + e^{-\tau\lambda} \Delta A_1.$$

Hence, we will denote the perturbed delay eigenvalue problem with matrices $A_0 + \Delta A_0$ and $A_1 + \Delta A_1$ by $D + \Delta D$. For a complex λ the *sign of* λ is defined as

$$\text{sign}(\lambda) = \begin{cases} \frac{\bar{\lambda}}{|\lambda|}, & \lambda \neq 0 \\ 0, & \lambda = 0. \end{cases}$$

In the following, $||\cdot||$ will stand for $||\cdot||_2$ and we will assume that the eigenvectors are normalized, i.e., $||x|| = 1$ for every eigenvector x.

4.3 Backward error

Definition 4.1. *Let $(\tilde{x}, \tilde{\lambda})$ be an approximate eigenpair of (4.1) and let $\tilde{x}$ be normalized. We define the* normwise backward error *of $(\tilde{x}, \tilde{\lambda})$ by*

$$\eta(\tilde{x}, \tilde{\lambda}) := \min\{\epsilon : (D(\tilde{\lambda}) + \Delta D(\tilde{\lambda}))\tilde{x} = 0, ||\Delta A_l|| \leq \epsilon ||E_l||, l = 0, 1\}, \tag{4.2}$$

The following theorem gives an explicit expression for $\eta(\tilde{x}, \tilde{\lambda})$.

Theorem 4.2. *For the normwise backward error $\eta(\tilde{x}, \tilde{\lambda})$ we have*

$$\eta(\tilde{x}, \tilde{\lambda}) = \frac{||r||}{\theta} \tag{4.3}$$

where $r := D(\tilde{\lambda})\tilde{x}$ *is the residual and*

$$\theta = ||E_0|| + |e^{-\tau\tilde{\lambda}}|||E_1||.$$

Proof:

$$\begin{aligned} D(\tilde{\lambda})\tilde{x} &= -\Delta D(\tilde{\lambda})\tilde{x} \\ ||D(\tilde{\lambda})\tilde{x}|| &= || - \Delta D(\tilde{\lambda})\tilde{x}|| \\ &\leq ||\Delta D(\tilde{\lambda})||||\tilde{x}|| = ||\Delta D(\tilde{\lambda})|| = ||\Delta A_0 + e^{-\tau\tilde{\lambda}}\Delta A_1|| \\ &\leq ||\Delta A_0|| + |e^{-\tau\tilde{\lambda}}|||\Delta A_1|| \\ &\leq \epsilon(||E_0|| + |e^{-\tau\tilde{\lambda}}|||E_1||) \end{aligned}$$

Thus we have a lower bound for the backward error

$$\eta(\tilde{x}, \tilde{\lambda}) \geq \frac{||D(\tilde{\lambda})\tilde{x}||}{\theta},$$

which is attained for

$$\Delta A_l = \frac{1}{\theta}||E_l||D(\tilde{\lambda})\tilde{x}, \qquad l = 0, 1.$$

□

If D is Hermitian then it is of interest to consider a backward error in which the perturbations ΔA_l are Hermitian. The backward error for the Hermitian delay eigenvalue problem can be defined as

$$\eta_H(\tilde{x}, \tilde{\lambda}) := \min\left\{\epsilon : (D(\tilde{\lambda}) + \Delta D(\tilde{\lambda}))\tilde{x} = 0, \Delta A_l^* = \Delta A_l, ||\Delta A_l|| \leq \epsilon||E_l||, l = 0, 1\right\}. \tag{4.4}$$

It is clear that $\eta_H(\tilde{x}, \tilde{\lambda}) \geq \eta(\tilde{x}, \tilde{\lambda})$ and that the optimal perturbations in (4.2) are not Hermitian in general. The next theorem shows that requiring the perturbations to respect the Hermitian structure in the A_l has no effect on the backward error, provided that $\tilde{\lambda}$ is real.

Theorem 4.3. *If* D *is Hermitian and* $\tilde{\lambda}$ *is real then*

$$\eta_H(\tilde{x}, \tilde{\lambda}) = \eta(\tilde{x}, \tilde{\lambda}). \tag{4.5}$$

Proof: Let $r = D(\tilde{\lambda})\tilde{x}$ be the residual of the pair $(\tilde{x}, \tilde{\lambda})$. We are looking for a Hermitian matrix S, such that $S\tilde{x} = -r$. We take $S = ||r||I$ if r is a negative multiple of $\tilde{x}$; otherwise we take $S = ||r||H$, where H is a Householder matrix that maps $\tilde{x}$ to $-r/||r||$. Such an H exists because $\tilde{x}^*S\tilde{x} = -\tilde{x}^*r$ is real, since $\tilde{\lambda}$ is real, and it is equal to $I - 2(w^*w)^{-1}ww^*$, where $w = \tilde{x} + r/||r||$.

Let ΔA_l be Hermitian matrices, defined by

$$\Delta A_l = \frac{1}{\theta}||E_l||H, \tag{4.6}$$

where $\theta = ||E_0|| + |e^{-\tau\tilde{\lambda}}|||E_1||$. It follows that $\Delta D(\tilde{\lambda}) = S$, and the first constraint in (4.4) is satisfied. Using (4.3), we get

$$||S|| = ||r|| \leq \eta(\tilde{x}, \tilde{\lambda})\theta.$$

From (4.6) we deduce that $\eta_H(\tilde{x}, \tilde{\lambda}) \leq \eta(\tilde{x}, \tilde{\lambda})$. Since $\eta_H(\tilde{x}, \tilde{\lambda}) \geq \eta(\tilde{x}, \tilde{\lambda})$ by definition, equality (4.5) must hold. □

If one is only interested in the approximate eigenvalue $\tilde{\lambda}$, then a more appropriate measure of the backward error may be

$$\eta(\tilde{\lambda}) := \min\left\{\eta(\tilde{x}, \tilde{\lambda}) \ : \ \tilde{x} \text{ normalized } \right\}.$$

Proposition 4.4.

$$\eta(\tilde{\lambda}) = \frac{1}{\theta}\sigma_{\min}(D(\tilde{\lambda})).$$

Proof: The result follows from Theorem 4.2 by using the equality

$$\min_{||x||=1} ||Ax|| = \sigma_{\min}(A).$$

□

4.4 Condition numbers

In this section, we assume that λ is a nonzero eigenvalue of a nonsingular delay eigenvalue problem D, with corresponding normalized right eigenvector x and left eigenvector y.

4.4.1 Eigenvalue condition number

Definition 4.5. *A normwise condition number of λ can be defined by*

$$\kappa(\lambda, D) := \limsup_{\epsilon \searrow 0} \left\{ \frac{||\Delta\lambda||}{\epsilon} : \ ((D+\Delta D)(\lambda+\Delta\lambda))\,(x+\Delta x) = 0, \right.$$
$$\left. ||\Delta A_l|| \leq \epsilon ||E_l||, l = 0, 1 \right\}. \quad (4.7)$$

The following theorem gives an explicit expression for $\kappa(\lambda, D)$.

Theorem 4.6. *The condition number $\kappa(\lambda, D)$ is given by*

$$\kappa(\lambda, D) = \frac{\theta ||y|| ||x||}{|\lambda| |y^* D'(\lambda) x|} \quad (4.8)$$

where $\theta = ||E_0|| + |e^{-\tau\lambda}| ||E_1||$ and x and y are the corresponding right and left eigenvectors, respectively.

Proof: Expanding the first constraint in the definition of the condition number we get

$$\Delta\lambda D'(\lambda)x + \Delta D(\lambda)x + D(\lambda)\Delta x = 0.$$

Premultiplying by y^* yields

$$\Delta\lambda y^* D'(\lambda)x + y^* \Delta D(\lambda)x = 0.$$

Considering λ to be a simple eigenvalue, $y^* D'(\lambda)x \neq 0$, one gets

$$\Delta\lambda = -\frac{y^* \Delta D(\lambda) x}{y^* D'(\lambda) x},$$

and so

$$\frac{|\Delta\lambda|}{\epsilon|\lambda|} \leq \frac{\theta ||y|| ||x||}{|\lambda| |y^* D'(\lambda) x|}.$$

Thus the expression in (4.8) is an upper bound for the condition number. Now consider the matrix

$$H = \frac{1}{||x|| ||y||} y x^*,$$

for which $||H|| = 1$ and $y^* H x = ||x|| ||y||$. Let

$$\Delta A_l = -\epsilon ||E_l|| H, l = 0, 1.$$

Then $||\Delta A_l|| = \epsilon||E_l||$ and

$$|y^*\Delta D(\lambda)x| = |y^*(-\epsilon\theta H)x| = \epsilon\theta|y^*Hx| = \epsilon\theta||x||||y||.$$

Hence, for these perturbation matrices, the upper bound can be attained

$$\frac{|\Delta\lambda|}{\epsilon|\lambda|} = \frac{\theta||y||||x||}{|\lambda||y^*P'(\lambda)x|}.$$

□

As for the backward error, if D is Hermitian, it is natural to restrict the perturbations ΔA_l in (4.8) to be Hermitian. We denote

$$\kappa_H(\lambda, D) := \limsup_{\epsilon\searrow 0}\left\{ \frac{||\Delta\lambda||}{\epsilon} : \ ((D+\Delta D)(\lambda+\Delta\lambda))\,(x+\Delta x) = 0, \right.$$
$$\left. \Delta A_l^* = \Delta A_l, ||\Delta A_l|| \le \epsilon||E_l||, l = 0, 1 \right\}. \quad (4.9)$$

Lemma 4.7. *If λ is a real eigenvalue of a Hermitian delay eigenvalue problem D, then*

$$\kappa_H(\lambda, D) = \kappa(\lambda, D).$$

Proof: For a Hermitian delay eigenvalue problem and real eigenvalue λ we can take $y = x$, and then the matrix H in the proof of Theorem 4.6 is Hermitian. It follows that the perturbations for which the bound is attained are also Hermitian. □

4.5 Pseudospectra

Another tool for the study of the sensitivity of the eigenvalues to perturbations are pseudospectra. They have been studied for the standard and generalized eigenproblem and for the polynomial eigenvalue problem [20] and multiparameter eigenvalue problem [11, Chapter 7] as well. In this section we extend the definition of the pseudospectrum to the delay eigenvalue problem. Note, that the pseudospectra of delay differential equations have already been studied in some papers, e.g., [15].

Definition 4.8. *We define the* ϵ-pseudospectrum of D *by*

$$\Lambda_\epsilon(D) = \{\lambda \in \mathbb{C} : \ D(\lambda) + \Delta D(\lambda) \text{ singular}, ||\Delta \mathrm{A_l}|| \le \epsilon||\mathrm{E_l}||, \mathrm{l} = 0, 1\}. \quad (4.10)$$

The following theorem gives explicit expressions for the pseudospectrum of D.

Theorem 4.9.

$$\begin{aligned}\Lambda_\epsilon(D) &= \{\lambda \in \mathbb{C} : \eta(\lambda) \leq \epsilon\} \\ &= \{\lambda \in \mathbb{C} : \sigma_{\min}(D(\lambda)) \leq \epsilon\theta\} \\ &= \{\lambda \in \mathbb{C} : ||D(\lambda)^{-1}|| \geq 1/(\epsilon\theta)\} \\ &= \{\lambda \in \mathbb{C} : \exists u, ||u|| = 1 \text{ with } ||\mathrm{D}(\lambda)\mathrm{u}|| \leq \epsilon\theta\}.\end{aligned}$$

Proof: The first equality follows readily from Definition 4.10. For the second equality Proposition 4.4 can be applied. The last two equalities follow from the identity

$$\min_{x \neq 0} \frac{||Ax||}{||x||} = ||A^{-1}||^{-1} = \sigma_{\min}(A),$$

with the convention that $||A^{-1}|| = \infty$ if A is singular. □

Next we present a numerical example of the previously discussed pseudospectra. We again select the problem of [12, Section 2.4.1] with the matrices A_0 and A_1 having dimension $N = 2$. Moreover, we consider relative perturbations, i.e., $E_0 = A_0$ and $E_1 = A_1$, and we draw the pseudospectra in Figure 4.1 by computing $\sigma_{\min}(D(\lambda))/\theta$ in all grid points by Matlab's `svd`. The size of the grid is 17×19.

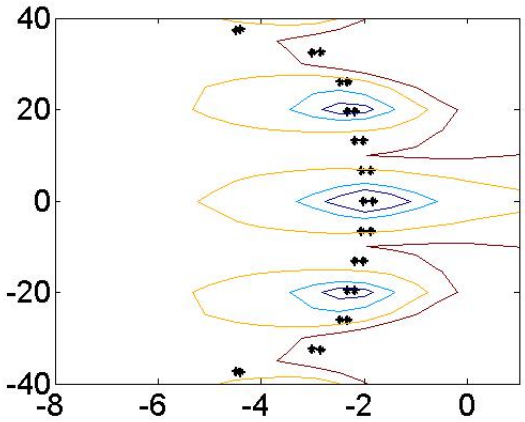

Figure 4.1: Pseudospectra for $\epsilon = 10^{-0.5}, 10^{-0.3}, 10^{-0.1}, 1$.

Chapter 5

Conclusions and Outlook

In the current thesis we have presented several rational approximation methods and consequently shown that they can be applied, more or less successfully, for solving nonlinear delay eigenvalue problems. The techniques that were found to be relatively effective are the scaled Padé approximation, the least squares approximation on a complex box, the CF method for the unit disk and the best approximation by Ellacott and Williams. As a comparison, we give the approximation errors for these methods in Table 5.1.

Padé	LS on box	CF on disk	Best rational		
$3.0 \cdot 10^{-4}$	$7.0 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$

Table 5.1: Errors for the rightmost eigenvalue by an $(8, 8)$ rational approximation of different types.

Moreover, we have shown that the Newton, Galerkin and minimal residual methods can all be used for refining an approximate eigenvalue of the given problem. Additionally, we have defined the backward error, condition number and pseudospectra of the delay eigenvalue problem.

As future steps, regarding this thesis, we mention the try-out of other rational approximation techniques, as well as some combinations of the ones discussed in the current paper, e.g., the use of a Padé approximant as initial guess for the least squares approximation on a complex box. Furthermore, we could introduce the so-called Jacobi–Davidson method for solving the delay eigenvalue problem, and comparing the results with the solutions of

some corresponding rational eigenvalue problems. In this thesis we were mainly focused on approximating the righmost eigenvalue of a given delay eigenvalue problem, but it would also be a good idea to try to give approximation errors for some other eigenvalues of the problem. Moreover, the perturbation theory of the delay eigenvalue problem could be developed further on.

Acknowledgments

I would like to thank, first of all, my supervisor, Michiel E. Hochstenbach, for guidance, attention and support throughout my work. The current thesis would not have been possible without his valuable comments and remarks. Moreover, I am particularly grateful to Elias Jarlebring for providing inspiration and giving constructive advices. And last, but not least, I would like to thank the members of the assessment committee: Lothar Reichel, Wil Schilders, and Sjoerd Rienstra for their time and patience to read and examine the current thesis and also for taking part as committee members at my thesis defense.

References

[1] Bellman, R., and Cooke, K. L. *Differential-Difference Equations.* Academic Press, 1963.

[2] Betcke, T. Optimal scaling of generalized and polynomial eigenvalue problems. `http://manchester.ac.uk/mims/eprints`, October 2007. MIMS Eprint.

[3] Boost C++ Libraries. `http://www.boost.org/doc/libs/1_38_0/libs/math/doc/sf_and_dist/html/math_toolkit/backgrounders/remez.html`.

[4] Burden, R. L., and Faires, J. D. *Numerical Analysis.* Brooks-Cole Publishing, 2004.

[5] Cody, W. J. A survey of practical rational and polynomial approximation of functions. *SIAM Review 12*, 3 (July 1970), 400–423.

[6] Ellacott, S., and Williams, J. Rational Chebyshev approximation in the complex plane. *Siam J. Numer. Anal. 13*, 3 (June 1976).

[7] Hale, J. K., and Verduyn Lunel, S. M. *Introduction to Functional Differential Equations*, vol. 99 of *Applied Mathematical Sciences.* Springer - Verlag, 1993.

[8] Higham, D. J., and Higham, N. J. Structured backward error and condition of generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl. 20*, 2 (1998), 493–512.

[9] Higham, N. J. Evaluating Padé approximants of the matrix logarithm. *SIAM J. Matrix Anal. Appl. 22*, 4 (2001), 1126–1135.

[10] Higham, N. J., Li, R.-C., and Tisseur, F. Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl. 29*, 4 (2007), 1218–1241.

[11] Hochstenbach, M. E. *Subspace Methods for Eigenvalue Problems.* Utrecht University, Department of Mathematics and Computer Science, 2003.

[12] Jarlebring, E. *The spectrum of delay-differential equations: numerical methods, stability and perturbation.* PhD thesis, Carl-Friedrich-Gauß-Fakultät, Technische Universität Carolo-Wilhelmina zu Braunschweig, 2008.

[13] Jarlebring, E., and Hochstenbach, M. E. Polynomial two-parameter eigenvalue problems and matrix pencil methods for stability of delay-differential equations. *Linear Algebra and its Applications 431*, 3-4 (July 2009), 369–380.

[14] Kincaid, D., and Cheney, W. *Numerical Analysis: Mathematics of Scientific Computing.* Brooks/Cole Publishers, Pacific Grove, CA., 1996.

[15] Michiels, W., Green, K., Wagenknecht, T., and Niculescu, S.-I. Pseudospectra and stability radii for analytic matrix functions with application to time delay systems. *Linear Algebra and its Applications 418*, 1 (2006), 315–335.

[16] Partington, J., and Mäkilä, P. M. Rational approximation of distributed-delay controllers. *International Journal of Control 78*, 16 (2005), 1295 – 1301.

[17] Schreiber, K. *Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals.* PhD thesis, TU Berlin, 2008.

[18] Tawfik, S. A. Minimax approximation and Remez algorithm. `http://www.mathworks.com/matlabcentral/fileexchange/8094`, July 2005.

[19] Tisseur, F. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra and its Applications*, 309 (2000), 339–361.

[20] Tisseur, F., and Higham, N. J. Structured pseudospectra for polynomial eigenvalue problems, with applications. *SIAM J. Matrix Anal. Appl. 23*, 1 (2001), 187–208.

[21] Trefethen, L. N. Near-circularity of the error curve in complex Chebyshev approximation. *Journal of Approximation Theory*, 31 (1981), 344–367.

[22] Trefethen, L. N. Rational Chebyshev approximation on the unit disk. *Numerische Mathematik*, 37 (1981), 297–320.

[23] Trefethen, L. N. *Spectral Methods in MATLAB*. SIAM, Philadelphia, 2000.

[24] Veidinger, L. On the numerical determination of the best approximations in the Chebyshev sense. *Numerische Mathematik 2* (1960), 99–105.

[25] Wolfram Mathworld. `http://mathworld.wolfram.com/`.