EFFICIENT ALGORITHMS FOR COMPUTING THE NEAREST POLYNOMIAL WITH CONSTRAINED ROOTS

By

Markus A. Hitz

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Science

Approved by the
Examining Committee:

_____
Erich Kaltofen, Thesis Adviser

_____
Joseph E. Flaherty, Thesis Adviser

_____
Wm. Randolph Franklin, Member

_____
Mukkai S. Krishnamoorthy, Member

_____
Franklin T. Luk, Member

_____
Robert McNaughton, Member

_____
Victor Pan, Member

Rensselaer Polytechnic Institute
Troy, New York

April 1998
(For Graduation May 1998)

EFFICIENT ALGORITHMS FOR COMPUTING THE NEAREST POLYNOMIAL WITH CONSTRAINED ROOTS

By

Markus A. Hitz

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Science

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Erich Kaltofen, Thesis Adviser
Joseph E. Flaherty, Thesis Adviser
Wm. Randolph Franklin, Member
Mukkai S. Krishnamoorthy, Member
Franklin T. Luk, Member
Robert McNaughton, Member
Victor Pan, Member

Rensselaer Polytechnic Institute
Troy, New York

April 1998
(For Graduation May 1998)

© Copyright 1998
by
Markus A. Hitz
All Rights Reserved

To the Memory of my Parents

Martha Hitz-Christen

(1923–1987)

Gotthold Hitz

(1916–1996)

CONTENTS

LIST OF FIGURES vi

Acknowledgement vii

Abstract viii

1. Introduction 1
   - 1.1 Computing with Uncertain Input Data 1
   - 1.2 Historical Overview 4
   - 1.3 Notation 6

2. Moving Polynomial Roots by Minimal Coefficient Perturbations 8
   - 2.1 Complex Coefficients 8
     - 2.1.1 Constraining a Single Root 8
     - 2.1.2 Constraining Several Roots 9
     - 2.1.3 Preserving Monicity 11
   - 2.2 Real Coefficients 11
   - 2.3 Computing Minimal Perturbations 13
     - 2.3.1 Minimal Perturbations in the l^2-Norm 14
     - 2.3.2 Minimal Perturbations in the l^∞-Norm and the l^1-Norm 16
   - 2.4 Weighted Norms and Other Constraints 18

3. Parametric Solutions 21
   - 3.1 Solutions for the l^2-Norm in Closed Form 21
   - 3.2 Constraining a Root Locus to a Curve 24
     - 3.2.1 Complex Coefficients 25
     - 3.2.2 Real Coefficients 26
   - 3.3 Approximate Greatest Common Divisors and Nearest Singular Polynomials 29
   - 3.4 Parametric Solutions in the l^∞-Norm 31

4. Computing the Radius of Stability 36
   - 4.1 Preliminaries 37
   - 4.2 The Nearest Unstable Polynomial – Complex Coefficients 38
   - 4.3 The Nearest Unstable Polynomial – Real Coefficients 41

5. Discussion 45
5.1 Conclusions 45
5.2 Future Directions 45
BIBLIOGRAPHY 48
APPENDIX 52
A. Approximate Solutions to Inconsistent Systems 53
A.1 Solutions in the l^{∞}-Norm 53
A.2 Solutions in the l^{1}-Norm 55

LIST OF FIGURES

1.1 Root displacements under a single-digit perturbation 3

2.1 Orthogonal projection in the l^2-norm 14

2.2 Projecting the coefficient vector onto the range of **M** in the l^∞-norm 15

2.3 Projecting the coefficient vector onto the range of **M** in the l^1-norm 17

2.4 Incorporating Equality Constraints 19

4.1 Hurwitz polynomial with complex coefficients 40

Acknowledgement

Many people contributed to the successful completion of my thesis research in one way or another, and I am grateful for this support. In the first place, I would like to thank my advisor, Erich Kaltofen. His expertise and critique was invaluable. I also would like to thank the National Science Foundation for its continued support through several research grants awarded to Erich Kaltofen. I thank all members of my doctoral committee for their service, especially Joe Flaherty, who acted as nominal chair during my defense to satisfy the requirements of the Graduate School at Rensselaer. For my most recent research, I would like to thank Lakshman for many inspiring discussions.

Finishing my thesis research at another university was both an interesting and sometimes cumbersome experience. I thank the staff of the Department of Computer Science at Rensselaer Polytechnic Institute, and the Mathematics Department at North Carolina State University for their generous help.

I thank my sister Brigitte and my brother Hanspeter, who took care of my affairs in Switzerland during all the time I spent in the United States.

Finally, my thanks go to my former colleagues, Tom Valente, Austin Lobo, and Angel Díaz, for taking part in many ventures, scholarly and otherwise.

Abstract

The location of polynomial roots is sensitive to perturbations of the coefficients. Continuous changes of the coefficients of a polynomial move the roots continuously. We consider the problem of finding the minimal perturbations to the coefficients to move one or several roots to given loci. We measure minimality in the Euclidean distance to the coefficient vector, as well as the maximal coefficient-wise change in absolute value (infinity norm), and in the Manhattan norm (l^1-norm). In the Euclidean norm the computational task reduces to a least squares problem, in the infinity norm and the l^1-norm it can be formulated as a linear program.

We can derive symbolic solutions in closed form for the Euclidean norm in the case of complex coefficients and a single complex root. Our new result is a formula for the minimum change in the infinity norm for the case of real coefficients and a single real root. Based on the principle of *parametric minimization* we develop hybrid symbolic-numeric algorithms to constrain one root of a complex or real polynomial to a curve in the complex plane.

As an application to robust control, we give a polynomial-time algorithm to compute the radius of stability in the Euclidean norm for a variety of stability domains.

CHAPTER 1
Introduction

Sensitivity of polynomials to coefficient perturbations is a major issue in algebraic and numerical computing. Small changes in the coefficients can cause large displacements of the roots. Wilkinson's "perfidious" polynomial is a frequently cited example for such behavior. We develop tools to determine how close, in terms of perturbations, a univariate polynomial is to other polynomials that have different root characteristics. In this chapter, we raise the basic problems that we are going to discuss. In a historical overview, we review recent results. Finally, we give a compilation of our notation that we use in the following chapters.

1.1 Computing with Uncertain Input Data

The precision of experimental or statistical data is inherently limited. Instruments can measure only a finite number of binary digits, and human observers are capable to read not more than a few decimals off an analog scale. Careful error analysis can determine a confidence interval based on a suitable distribution for the data points. If such data is processed by mathematical software, the unwary user is sometimes confronted with seemingly random results.

The following polynomial over the reals might well have been obtained from a scientific experiment:

$$f(x) = x^2 - 1.00010x + 0.25005\,.$$

It is assumed to be monic, while the other two coefficients are given in the precision shown here. If we compute the roots of f up to six significant decimal digits, we see that f has the two real roots:

$$x_1 = 0.50000 \quad \text{and} \quad x_2 = 0.50010\,.$$

However, if the last decimal of the coefficients is uncertain, could have ended up with the

slightly *perturbed* polynomial

$$\tilde{f}(x) = x^2 - 1.00010x + 0.25006$$

that now has a pair of conjugate complex roots, namely

$$z_{1,2} = 0.50005 \pm 0.00316\mathbf{i}.$$

In figure 1.1, we plotted the root locations of the family of polynomials that we get if we change the last digit of the second coefficient in small steps from 5 to 6. Obviously there is a polynomial in this family that has a double root near $x = 0.50005$. It is well-known that if f is close to such a *singular* polynomial, root-finding is an ill-conditioned problem (see Demmel [1987]). The sensitivity to coefficient perturbations is not depending on the kind of arithmetic, floating point or exact rational, that we employ to determine the roots; it is the problem itself that is ill-posed.

Assuming that only the leading digits are certified, and the last digit of the coefficients of f is uncertain, we arrive at the crucial question: which answer, two real roots, one double (real) root, or two conjugate complex roots, is the right one? If we are unable to increase the precision of the given coefficients then we have to accept the results of our computation as coexistent solutions to the root-finding problem for the *family* of polynomials defined by the range of the uncertainty. In the following chapters, we will use norms of the coefficient vector to measure the overall size of the uncertainty.

For univariate polynomials over the real or complex numbers, we can ask a variety of questions that are related to the root-finding problem in presence of uncertainties. We propose to base these problems on the following three fundamental questions:

Q1: Given a polynomial f with real or complex coefficients and a real or complex number α. Is f "near" another polynomial $\tilde{f}$ such that $\tilde{f}(\alpha) = 0$?

Q2: Given two real or complex polynomials f and g with $\mathrm{GCD}(f,g) = 1$. Are f and g "near" polynomials $\tilde{f}$ and $\tilde{g}$ that have a non-trivial greatest common divisor?

Q3: Given a squarefree polynomial f with real or complex coefficients. Is f "near" a singular polynomial $\tilde{f}$?

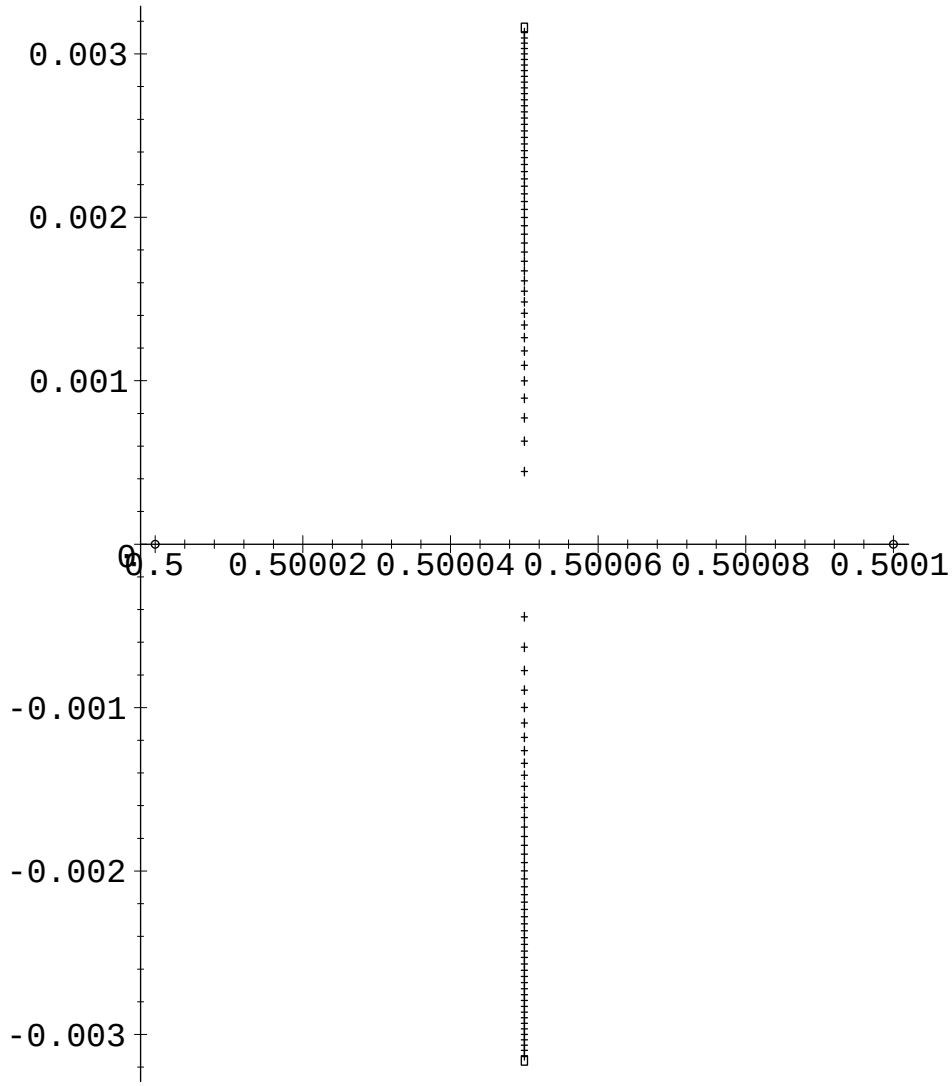

Figure 1.1: Root displacements under a single-digit perturbation

Much of the research in this area was focused on questions Q2 and Q3 so far, where Q3 was considered to be the most basic at all. In the following, we want to study Q1 in more depth, and we want to show that Q1 is actually the primordial problem.

1.2 Historical Overview

Motivated by the "ill condition" of the root-finding problem, question Q3 was subject of a thorough investigation already in the seventies. In his thesis, Hough [1977] was able to develop methods, based on Langrange multipliers, for computing nearest singular polynomials. Subsequently, some of his results were used by Demmel [1987] in his more general approach of finding condition numbers for several problems, showing that there is a common underlying structure in the mathematical formulation of the "distance to the set of ill-posed problems". However, he was more interested in deriving bounds than actually computing nearest singular objects. From the existential theory of the reals by Renegar [1992b], we know that our (and many similar) problems, have at least solutions that can be computed in single exponential time. In Karmarkar and Lakshman [1996], question Q3 was shown to be exactly solvable in polynomial time, if one uses the Euclidean norm.

The majority of the literature however, is devoted to the study of question Q2. Many methods, most of them applying heuristics, were devised to find "approximate GCDs", "near-GCDs", or "ε-GCDs". Schönhage [1985] uses the term "quasi-GCD"to define an asymptotically close GCD of two relatively prime polynomials. The assumption that the polynomial coefficients can be retrieved in arbitrary precision by making oracle calls, puts his method in opposition to our model of uncertain input data. Noda and Sasaki [1991] use an adaptive version of the Euclidean algorithm to compute approximate GCDs. As their algorithm uses local information for scaling, they are not guaranteed to achieve a global minimum. The method of Karcanias and Mitrouli [1992] and Karcanias and Mitrouli [1994] is formulated in the language of control theory. Originally, it was designed to compute the GCD of several polynomials numerically. It turned out that it was also capable of computing approximate GCDs of relatively prime polynomials. It can be shown that Karcanias and Mitrouli [1994] is a variant of Lazard [1981]'s more general method for solving systems of algebraic equations. Another numerical approach is based on the singular value decomposition (SVD). In Corless et al. [1995], it is used to deter-

mine a numerical rank of the Sylvester matrix, and to derive an approximate GCD from the perturbed matrix that is obtained by lowering the original rank. Because the structure of the Sylvester matrix is destroyed during this process, the approximate GCD cannot be optimal. However, the method allows to find an upper bound for the degree of approximate GCDs of *highest degree*. Recently, Emiris et al. [1997] developed a gap theory to improve the SVD method. Hribernig and Stetter [1997] use another numerical variant of the Euclidean algorithm to compute "near-GCDs". They are able to give a lower bound on the degree of the nearest GCD of highest degree.

Victor Pan proposes a significantly different approach, to avoid large root displacements introduced by coefficient perturbations that are applied to the original polynomials in the process of computing approximate GCDs. The most comprehensive description of his method is Pan [1998]. The algorithm computes all zeroes of the given polynomials, and then tries to find an optimal match in a bipartite graph. The selected roots (one of each polynomial) are then contracted to their mean value, and the perturbed coefficients are computed by backward-expansion of the polynomials.

All methods described so far are truly approximate. It was Karmarkar and Lakshman [1996] to devise the first algorithm that provably computes the global optimum for the nearest approximate GCD of two polynomials in the l^2-norm. Their method uses *parametric minimization*, and will be further explored in the following chapters. In Karmarkar and Lakshman [1996] a quadratic form had to be minimized to obtain a symbolic minimum of the norm of perturbations. This step is equivalent to solving a least squares problem. We will apply this simpler formulation in chapter 2, and extend it to other norms. Karmarkar and Lakshman [1997] uses Langrange multipliers to derive solutions for the symbolic minimum in closed form. In chapter 3, we will use projection onto a one-dimensional subspace to simplify this derivation. Whereas the global optimum for the nearest approximate GCD in the l^2-norm can now be found in polynomial-time, the problem of finding an approximate GCD of highest degree still results in single-exponential running time (see Karmarkar and Lakshman [1997]).

Although the classic text by Marden [1966] contains some material on perturbing polynomial coefficients, it does not provide us with an answer to Q1. To our knowledge, Q1 has not been directly addressed in the more recent literature. However, some problems

that are related to Q1 have been extensively investigated in control theory, especially the clustering of roots in an open domain of the complex plane. The latter is referred to as *domain stability* of a polynomial. *Robust* stability requires that all roots of a polynomial stay in a given domain even under perturbation of its coefficients. A remarkable result for the special case of the left half-plane (*Hurwitz* stability) is Kharitonov [1978a]. It reduces the continuous problem of establishing Hurwitz stability for an entire family of real interval polynomials to a discrete problem of testing only *four* special members of the given polynomial family. One year later, the theorem was extended to complex interval polynomials in Kharitonov [1979]; the test set consists of eight "corner" polynomials in that case. Unfortunately, Kharitonov's theorem cannot be extended to many other domains (see Rantzer [1992]). Nevertheless, it had a considerable impact on research in robust control. Many Kharitonov-like tests have been developed in the meantime, for a variety of domains and norms (see Tempo [1989], Barmish and Kang [1993]).

In an effort to arrive at a method for perturbing various objects while preserving their structure, Doyle [1982] introduced the notion of *structured singular values*. In terms of computations the method turned out to be extremely costly in the general case, and soon the focus shifted to computing upper bounds instead of the actual decomposition. One can easily find a nearest singular matrix by perturbing a given regular matrix if we do not enforce any structure (see Kahan [1966]). However, the landmark result by Poljak and Rohn [1993] shows that computing the distance to the nearest singular matrix becomes an NP-hard problem when a component-wise infinity norm is used (see also Demmel [1992]). As a consequence, the general form of the structured singular value problem is NP-hard as well. For the rather special class of *rank-one* problems, there exist efficient algorithms for computing structured singular values. They can be successfully applied to the stability problem for polynomials (see Qiu and Davison [1989], Chen et al. [1994a,b]). More specific literature with respect to the stability problem will be discussed at the beginning of chapter 4.

1.3 Notation

In the following, $\mathbb{C}[z]$ and $\mathbb{R}[x]$ are the rings of univariate polynomials over the complex numbers and real numbers, respectively. Polynomials are assumed to be of generic

degree n, unless stated otherwise. The roots are denoted by Greek letters. When we use the norm expression $\|f\|$ for a polynomial f it is meant to be the norm of the coefficient vector of f.

Matrices and vectors are type-set in bold; a bold $\mathbf{i}$ stands for the imaginary unit of the complex numbers. What type of vector norm, $\|.\|$ stands for, is depending on the context. In chapter 3 it is assumed to represent the Euclidean norm (l^2 norm) for the major part.

The operator $[.]^{tr}$ is the transposition applied to vectors and matrices, whereas the Hermite transposition (over the complex numbers) is denoted by an asterisk $[.]^*$. We use the standard definition for the inner product of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$:

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{y}^* \mathbf{x} = \sum_{k=1}^{n} \mathbf{x}_k \overline{\mathbf{y}}_k \,.$$

The complex conjugate of $c = a + b\mathbf{i} \in \mathbb{C}$ is marked by a bar: $\overline{c} = a - b\mathbf{i}$. The absolute value of a complex number is denoted by $|c|$, where $|c|^2 = \overline{c}c = a^2 + b^2$.

CHAPTER 2
Moving Polynomial Roots by Minimal Coefficient Perturbations

In the following we take a given univariate polynomial and perturb its coefficients such that one or several roots move to prescribed locations in the complex plane under the condition that the induced perturbation be minimal. Naturally, the perturbed polynomial has the same or a smaller degree then the given polynomial. More succinctly, we want to compute the coefficients of the or a *nearest* perturbed polynomial that has roots at the given locations, where nearness is measured by a certain norm of the coefficient vector. We first state the problem for a generic vector norm $\|.\|$, then we discuss solution strategies for the l^1, l^2, and l^∞-norm. Later on, we briefly look at weighted norms, and how *linear* constraints among the perturbed coefficients can be incorporated.

2.1 Complex Coefficients

First, we state the problem for a *single* root α with prescribed locus in the complex plane. This special case will be the basis for the formulas in the next chapter, as well as for our application to control theory.

2.1.1 Constraining a Single Root

Problem 1 *Given* $f \in \mathbb{C}[z]$ *with* $\deg(f) = n$. *For the complex indeterminate* α, *find* $\tilde{f} \in \mathbb{C}[z]$, *such that*

$$\tilde{f}(\alpha) = 0\,, \quad \text{and} \quad \|f - \tilde{f}\| = \min\,.$$

Now let

$$f \in \mathbb{C}[z]\,, \quad f(z) = \sum_{k=0}^{n} a_k z^k\,.$$

We define the perturbed, polynomial $\tilde{f} \in \mathbb{C}[z]$ such that $\tilde{f}(\alpha) = 0$:

$$\begin{aligned}
\tilde{f}(z) &= (z-\alpha)\sum_{k=0}^{n-1} u_k z^k \\
&= u_{n-1}z^n + (u_{n-2}-\alpha)z^{n-1} + (u_{n-3}-\alpha u_{n-2})z^{n-2} + \cdots + (u_0 - \alpha u_1)z - \alpha u_0 \\
&= u_{n-1}z^n + \sum_{k=1}^{n-1}(u_{k-1}-\alpha u_k)z^k - \alpha u_0\,,
\end{aligned}$$

where $u_k \in \mathbb{C}$. Furthermore, we define the perturbation $p = f - \tilde{f}$, a polynomial of degree n or less. Next, we formulate the problem in terms of linear algebra. We define the vectors

$$\begin{aligned}
\mathbf{b} &= [a_0, \ldots, a_{n-1}, a_n]^{tr} \in \mathbb{C}^{n+1}\,, \quad \text{and} \\
\mathbf{w} &= [u_0, \ldots, u_{n-1}]^{tr} \in \mathbb{C}^{n}\,.
\end{aligned}$$

The vector $\|\delta\| \in \mathbb{C}^n$ of perturbations shall contain the coefficients of the polynomial p. In order to compute $\|\delta\|$, we have to solve the following optimization problem:

$$\|\delta\| = \min_{\mathbf{w}\in\mathbb{C}^n} \|\mathbf{M}\mathbf{w} - \mathbf{b}\|\,, \tag{2.1}$$

where the rectangular matrix

$$\mathbf{M} = \begin{bmatrix} -\alpha & & & & 0 \\ 1 & -\alpha & & & \\ & \ddots & \ddots & & \\ 0 & & 1 & -\alpha & \\ & & & 1 & \end{bmatrix} \in \mathbb{C}^{(n+1)\times n} \tag{2.2}$$

represents the polynomial multiplication operator for $(z-\alpha)$.

2.1.2 Constraining Several Roots

In an analogous fashion, we formulate the problem for m roots, where $m < n$:

Problem 2 *Given* $f \in \mathbb{C}[z]$ *with* $\deg(f) = n$, $\alpha_1, \ldots, \alpha_m \in \mathbb{C}$, *and* $m < n$. *Find* $\tilde{f} \in \mathbb{C}[z]$, *such that*

$$\tilde{f}(\alpha_k) = 0 \quad \text{for} \quad 1 \le k \le m\,, \quad \text{and} \quad \|f - \tilde{f}\| = \min\,.$$

This time, we have to multiply by the polynomial factor

$$\begin{aligned}\prod_{k=1}^{m}(z-\alpha_k) &= z^m - (\alpha_1 + \cdots + \alpha_m)z^{m-1} + \cdots - \cdots + (-1)^m \alpha_1 \cdots \alpha_m \\ &= \sum_{k=0}^{m}(-1)^k \sigma_k z^{m-k}\,,\end{aligned}$$

where the σ_k are the elementary symmetric functions in $\alpha_1, \ldots, \alpha_m$, and $\sigma_0 = 1$.

Maintaining the notation from above, we arrive at the optimization problem

$$\|\delta\| = \min_{\mathbf{w}_m \in \mathbb{C}^{n-m+1}} \|\mathbf{M}_m \mathbf{w}_m - \mathbf{b}\|\,, \tag{2.3}$$

where now the coefficient vector $\mathbf{w}_m$ of the co-factor has $n - m + 1$ elements,

$$\mathbf{w}_m = [u_0, \ldots, u_{n-m}]^{tr} \in \mathbb{C}^{(n-m+1)}\,,$$

and the matrix

$$\mathbf{M}_m = \begin{bmatrix} (-1)^m\sigma_m & & & 0 \\ \vdots & (-1)^m\sigma_m & & \\ -\sigma_1 & \vdots & \ddots & \\ 1 & -\sigma_1 & & (-1)^m\sigma_m \\ & 1 & \ddots & \vdots \\ 0 & & \ddots & -\sigma_1 \\ & & & 1 \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n-m+1)} \tag{2.4}$$

represents the multiplication operator for

$$\prod_{k=1}^{m}(z-\alpha_k)\,.$$

2.1.3 Preserving Monicity

Sometimes it is desirable to restrict oneself to monic polynomials. As we will see later on, the resulting minimal perturbations contain the leading coefficient (or its absolute value) as scaling factor. To avoid "scaling", one can use monic polynomials only. In addition, the degree of the perturbed polynomial will be the same as the given polynomial, thus generating a family of *fixed-degree* polynomials. An important example is the algorithm for computing approximate greatest common divisors in the l^2-norm by Karmarkar and Lakshman [1996, 1997].

We demonstrate the method for the case of a single root, the generalization to several roots is obvious. Problem 1 becomes:

Problem 3 *Given* $f \in \mathbb{C}[z]$ *monic,* $\deg(f) = n$*, and* $\alpha \in \mathbb{C}$*. Find* $\tilde{f} \in \mathbb{C}[z]$*, monic, such that*

$$\tilde{f}(\alpha) = 0\,, \quad \text{and} \quad \|f - \tilde{f}\| = \min\,.$$

Due to the monicity constraint the leading coefficients are $a_n = 1$ and $u_{n-1} = 1$, and are not part of the equation. The optimization problem in terms of linear algebra

$$\|\delta\| = \min_{\mathbf{w}^{(1)} \in \mathbb{C}^{n-1}} \|\mathbf{M}^{(1)}\mathbf{w}^{(1)} - \mathbf{b}^{(1)}\|\,, \tag{2.5}$$

is made up of

$$\begin{aligned} \mathbf{b}^{(1)} &= [a_0, \ldots, a_{n-2}, a_{n-1} + \alpha]^{tr} \in \mathbb{C}^n\,, \\ \mathbf{w}^{(1)} &= [u_0, \ldots, u_{n-2}]^{tr} \in \mathbb{C}^{n-1}\,, \end{aligned}$$

and the matrix $\mathbf{M}^{(1)} \in \mathbb{C}^{n\times(n-1)}$, which is the $n \times (n-1)$ leading minor of matrix $\mathbf{M}$ in (2.2). In the last element of the constant vector $\mathbf{b}^{(1)}$, we have to add $u_{n-1}\alpha = \alpha$ to account for the deletion of the last row and last column of $\mathbf{M}$.

2.2 Real Coefficients

Especially for sensitivity analysis, it is important that a given polynomial $f \in \mathbb{R}[x]$ retains real coefficients under perturbation. This constraint leaves us with two choices for the roots whose loci we want to constrain: they have to be either real, or they have

to belong to a pair of conjugate complex numbers. The case of real coefficients and *real* roots is identical to the cases of complex coefficients and complex roots described in the previous section, except that all variables would now represent real numbers.

The case of a pair of conjugate complex roots can be stated as follows:

Problem 4 *Given* $f \in \mathbb{R}[x]$ *and* $\alpha \in \mathbb{C}$. *Find* $\tilde{f} \in \mathbb{R}[x]$ *such that*

$$\tilde{f}(\alpha) = 0\,, \quad \tilde{f}(\overline{\alpha}) = 0\,, \quad \text{and} \quad \|f - \tilde{f}\| = \min\,.$$

Let $\alpha = a + b\,\mathbf{i}$, where $a \in \mathbb{R}$ and $b \in \mathbb{R}$, then the perturbed polynomial $\tilde{f} \in \mathbb{R}[x]$ has a factor

$$\begin{aligned}(x-\alpha)(x-\overline{\alpha}) &= x^2 - 2ax + a^2 + b^2 \\ &= x^2 + sx + q\,,\end{aligned}$$

where

$$s, q \in \mathbb{R}\,,\ q \geq 0\,, \quad \text{and} \quad a = -s/2\,, \quad b^2 = q - s^2/4\,.$$

Analogous to (2.1), we can formulate the minimization problem as:

$$\|\delta\| = \min_{\mathbf{w} \in \mathbb{R}^{n-1}} \|\mathbf{M}_r \mathbf{w}_r - \mathbf{b}\|\,, \tag{2.6}$$

where

$$\begin{aligned}
f(x) &= a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0\,, \quad n \geq 3\,, \\
\tilde{f}(x) &= u_{n-2} x^n + (u_{n-3} + s u_{n-2}) x^{n-1} + (u_{n-4} + s u_{n-3} + q u_{n-2}) x^{n-2} \\
&\quad + (u_{n-5} + s u_{n-4} + q u_{n-3}) x^{n-3} + \cdots \\
&\quad + (u_0 + s u_1 + q u_2) x^2 + (s u_0 + q u_1) x + q u_0\,, \\
\mathbf{b} &= [a_0, \ldots, a_{n-1}, a_n]^{tr} \in \mathbb{R}^{n+1}\,, \\
\mathbf{w}_r &= [u_0, \ldots, u_{n-2}]^{tr} \in \mathbb{R}^{n-1}\,,
\end{aligned}$$

and

$$\mathbf{M}_r = \begin{bmatrix} q & & & & 0 \\ s & q & & & \\ 1 & s & q & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & s & q \\ 0 & & & 1 & s \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n-1)} . \tag{2.7}$$

For symbolic derivations, the information $q \geq 0$ should be incorporated as an additional constraint. This can be accomplished by either directly adding the constraint $q \geq 0$ in a linear program, or by using an appropriate directive in a computer algebra system. For example, MAPLE provides the user with the `assume` function to define properties of a symbolic variable.

2.3 Computing Minimal Perturbations

The minimization problems described in the previous sections lead to over-determined linear systems of equations. In the following we refer to the matrix $\mathbf{M}$, and the vectors $\mathbf{w}$ and $\mathbf{b}$ as representatives for any of the special cases (2.1), (2.3), (2.5), and (2.6). A minimally perturbed coefficient vector for $\tilde{f}$ can be obtained by projecting the constant vector $\mathbf{b}$ onto the range of $\mathbf{M}$ in some specific norm. First we state, in a lemma, that the dimension of the column space of $\mathbf{M}$ is equal to the number of components of $\mathbf{w}$:

Lemma 1 *For any given root locus (loci)* α*, the matrix* $\mathbf{M}$ *has full rank.*

PROOF: The column vectors of the matrices $\mathbf{M}$, that we constructed for the minimization problems of the previous sections, are linearly independent for any choice of values for the indeterminate(s) α. If $\alpha = 0$ then the columns become unit vectors. □

For the case (2.1) of a single root, the range of $\mathbf{M}$ is an n-dimensional hyperplane in $(n+1)$-dimensional coefficient space.

The following lemma, although being trivial, is an important tool to check the validity of symbolic solutions for the special case when $\alpha = 0$:

Lemma 2 *Let $f \in \mathbb{C}[z]$ or $f \in \mathbb{R}[x]$, $deg(f) = n$. The minimal perturbation δ to transform f into $\tilde{f}$, such that $\tilde{f}(0) = 0$, is given by $\|\delta\| = |a_0|$. For the l^2 and the l^1 norm, the components of δ are:*

$$\delta_0 = a_0\,, \quad \text{and} \quad \delta_k = 0\,, \quad \text{for} \quad 1 \le k \le n\,.$$

PROOF: In order to have $\alpha = 0$ as a root, the 0th coefficient has to vanish, i.e., $\|\delta\|$ has to be at least $|a_0|$. In the l^2 and the l^1 norm, any perturbation of another coefficient would only increase $\|\delta\|$. □

Depending on the norm that we are applying, we have to use different solution methods (see Golub and Van Loan [1983]). In the following sub-sections we discuss the strategies for the l^2, the l^∞, and the l^1- norm. Finally, we explain how to extend these methods to weighted norms, and how one can incorporate linear constraints among the perturbed coefficients.

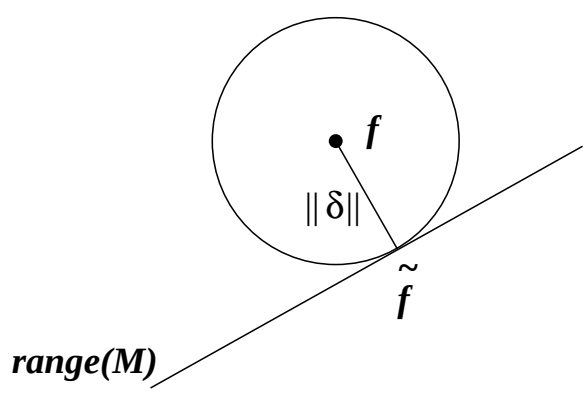

Figure 2.1: Orthogonal projection in the l^2-norm

2.3.1 Minimal Perturbations in the l^2-Norm

Because of lemma 1, each minimization problem has a unique *least squares* solution in the l^2-norm. Figure 2.1 illustrates the principal geometric relationship for a two-dimensional, real coefficient space, and a single (real) constrained root. In general,

the l^2-norm solution is the *orthogonal* projection onto the range of **M**, while the norm of the perturbation is the radius of the hyper-sphere to which the range of **M** is tangent at the solution point in coefficient space.

Both, computer algebra systems and numerical software packages feature built-in least squares solvers. However, computer algebra systems can also derive *symbolic* least squares solutions in the indeterminate(s) α. In the next chapter, we will derive formulas in closed form for the case of a single complex root. They are also the solution in the case of real coefficients and a single real root.

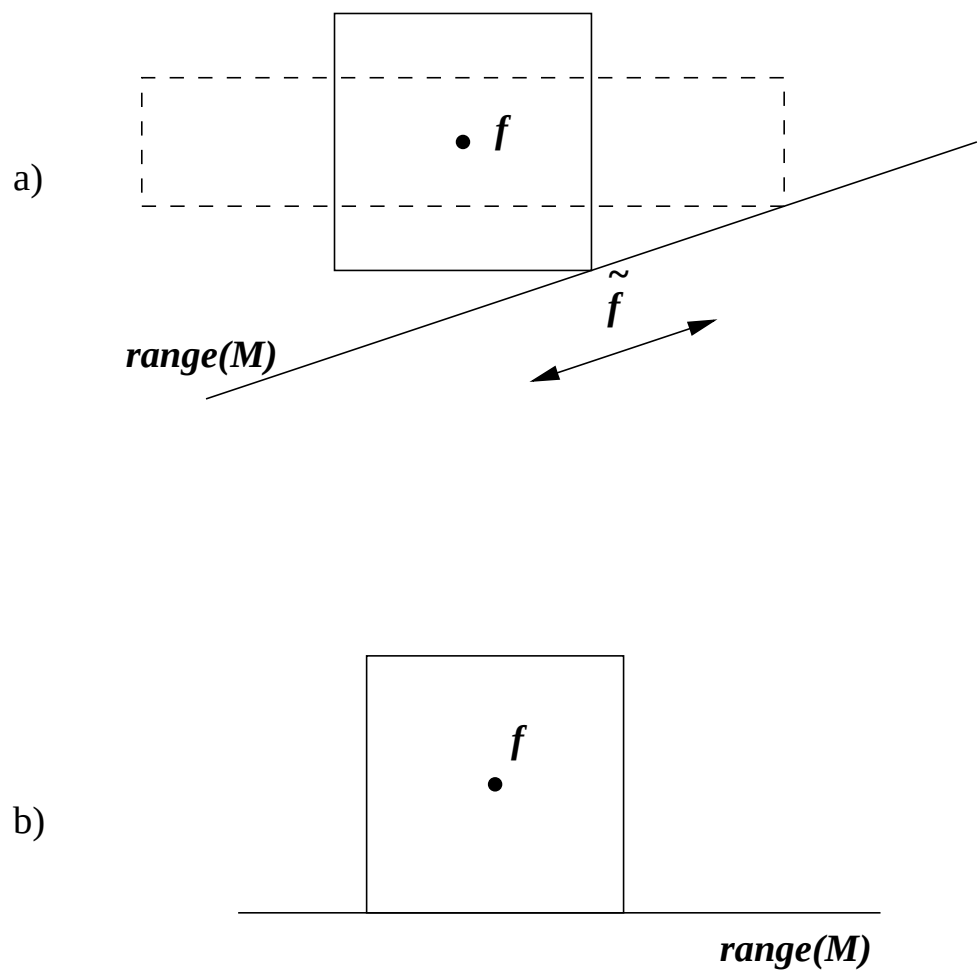

Figure 2.2: Projecting the coefficient vector onto the range of M in the l^∞-norm

2.3.2 Minimal Perturbations in the l^∞-Norm and the l^1-Norm

There exist many efficient algorithms for minimization in the infinity norm and the l^1-norm. Golub and Van Loan [1983] contains several references, a more recent article is Coleman and Li [1992a], or its l^1-counterpart Coleman and Li [1992b]. All these methods have in common that they use highly optimized numerical techniques, ultimately achieving quadratic convergence like in Newton iteration.

To better understand the real nature of l^∞ and l^1 minimization problems, we have to go back to the underlying basic approach, namely the formulation as a linear program. An early reference for the case of the infinity norm is Stiefel [1960]. Figure 2.2 illustrates the geometric relationship for a two-dimensional real coefficient space and a single real root. If $\alpha \neq 0$ then the perturbation (or residual) in the l^∞-norm is minimal when all components have the same absolute value $||\delta||$, as the l^∞-norm is defined by

$$||\delta||_\infty = \max_k |\delta_k| \,.$$

If we would move away from the optimal projection point then at least the absolute value of one component would increase, as indicated by the dashed "box" in part a) of the picture. The objective function of the linear program is $||\delta||_\infty$ whereas each component of the perturbation vector contributes two constraints, one for the positive, and one for the negative direction. Part b) shows that if one edge, or in the general case, one hyper-facet, is parallel to the range of $\mathbf{M}$ then we have a continuous set of minima. Also in the case when $\alpha = 0$ we have infinitely many solutions (see lemma 2), because we only have to make the constant coefficient vanish, the other coefficients can be chosen freely within the range of $||\delta||$. Similar statements can be made in the case of several roots when all or some are set to zero. This exception is caused by one (or several) rows in $\mathbf{M}$ becoming zero vectors.

The picture for the l^1-norm is shown in figure 2.3. Again, there is the possibility for an infinite number of minimal solutions, as indicated in part b) of figure 2.3.

We supply the linear programs in the appendix, together with simple MAPLE procedures that we used for experiments. They make function calls to the simplex algorithm which has exponential running time in the worst case, and therefore are not efficient. However by applying *interior point* methods for solving linear programs, like Karmarkar

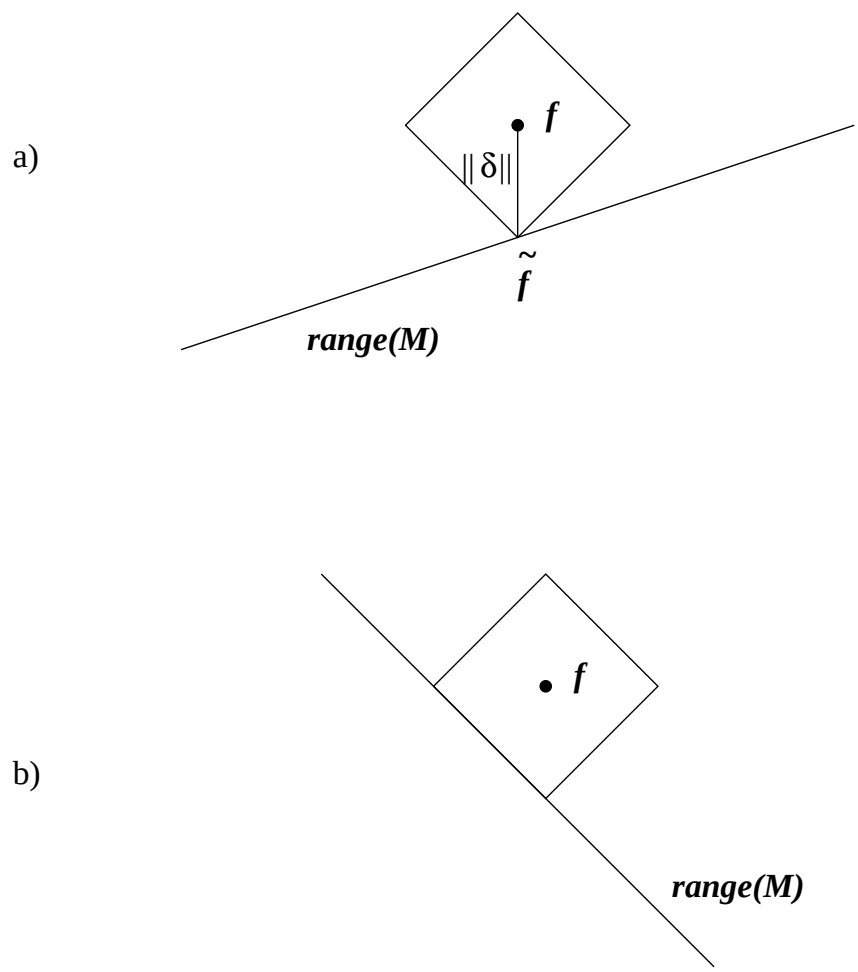

Figure 2.3: Projecting the coefficient vector onto the range of M in the l^1-norm

[1984] or Renegar [1988], we would arrive at a polynomial-time algorithm for any input. Numerically one can then expect linear convergence (see Coleman and Li [1992b]).

We have to make clear that we can only solve problems over the real numbers directly. However in the complex case, we can apply a well known technique (see, e.g., Young and Gregory [1973]) to separate real and imaginary parts. We demonstrate the decomposition for the case of a single complex root $\alpha = a + b\,\mathbf{i}\,,\ a, b \in \mathbb{R}$:
Let the constant vector in (2.1)

$$\mathbf{w} = [u_0 + v_0\,\mathbf{i}, \ldots, u_{n-2} + v_{n-2}\,\mathbf{i}]^{tr}\,,$$

and the matrix $\mathbf{M} = \mathbf{A} + \mathbf{i}\,\mathbf{B}$, where

$$\mathbf{A} = \begin{bmatrix} -a & & & & 0 \\ 1 & -a & & & \\ & \ddots & \ddots & & \\ 0 & & 1 & -a & \\ & & & 1 & \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} -b & & & & 0 \\ 0 & -b & & & \\ & \ddots & \ddots & & \\ 0 & & 0 & -b & \\ & & & 0 & \end{bmatrix}$$

are real $n \times (n-1)$ matrices. If we define

$$\mathbf{u} = [u_0, \ldots, u_{n-2}]^{tr} \text{ and } \quad \mathbf{v} = [v_0, \ldots, v_{n-2}]^{tr},$$

then we can replace $\mathbf{Mw}$ by the product of a block-matrix times a stacked vector

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix},$$

and set up the minimization problem accordingly.

2.4 Weighted Norms and Other Constraints

Sometimes it is useful to introduce positive weights to allow some coefficients to have larger or smaller perturbations than the others. A weighted norm applies a positive real factor to each component. E.g., a weighted l^2-norm of a complex vector $\mathbf{x}$ is given by

$$\|\mathbf{x}\|_w = \left(\sum_{k=1}^{n} w_k \overline{x_k} x_k \right)^{1/2},$$

where $w_k \in \mathbb{R}$ and $w_k > 0$ for $1 \leq k \leq n$.

For the l^∞ and the l^1 norm, the weights can be incorporated into the formulation of the linear program to find solutions to our minimization problems.

Similarly, least squares problems can be weighted (see GVL83, sec. 6.3). Typically, the weights are contained in a matrix $\mathbf{D} = \mathrm{diag}(w_k)$, and the least squares problem is to determine:

$$\min_{\mathbf{w} \in \mathbb{C}^n} \|\mathbf{D}(\mathbf{Mw} - \mathbf{b})\|_2 .$$

For the case of complex coefficients and a single root α, Karmarkar and Lakshman [1997] give a solution in closed form in a weighted l^2-norm (see also Hitz and Kaltofen [1998]).

A further generalization is given, if we impose *linear equality constraints* on the coefficients of the perturbed polynomial. The principal situation in this case is depicted in figure 2.4: we now have to project the constant vector onto the intersection of the

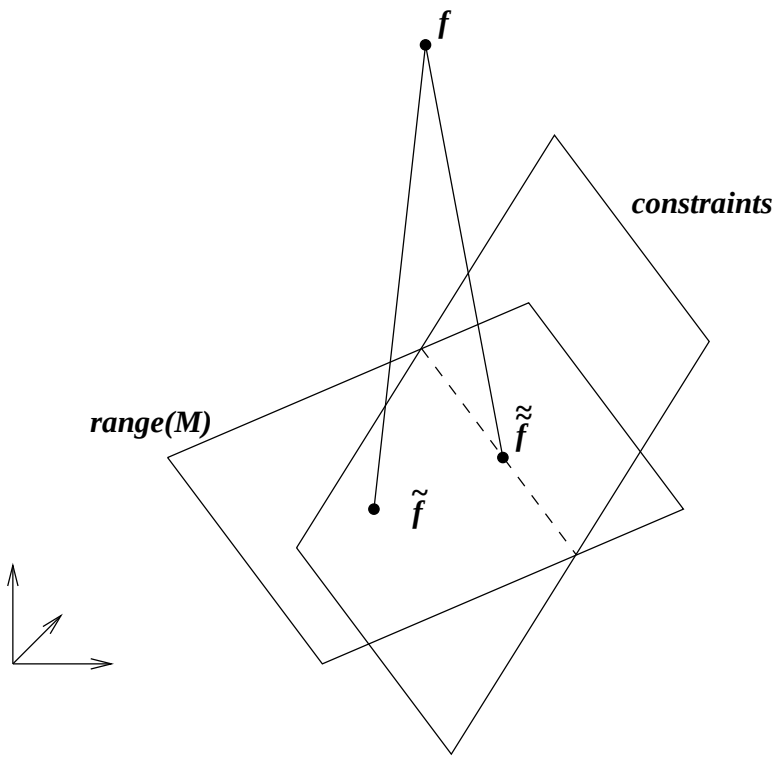

Figure 2.4: Incorporating Equality Constraints

range of **M** with the subspace spanned by the linear constraints. The solution under constraints is denoted by $\tilde{\tilde{f}}$. Obviously, not any choice of constraints will lead to a non-empty intersection of the two subspaces, and there might be no solution.

The constraints can be directly added to the linear program in the case of the l^∞ and the l^1 norm.

Methods that compute an orthonormal basis for the intersection of two subspaces are described in Golub and Van Loan [1983]. They are based on the singular value decomposition. From an orthonormal basis, on can construct a matrix **P** that projects any

vector **b** onto the intersection (see Golub and Van Loan [1983], sec. 2.4).

In the l^2, *inequality* constraints seem to be more difficult to incorporate, whereas in the l^∞ and the l^1 norm, they can be added to the linear program again. Methods that solve constraint least squares problems are described in Golub and Van Loan [1983], sec. 12.1.

Constraints could allow us to treat conditions like monicity or sparsity in a uniform way.

CHAPTER 3
Parametric Solutions

The methods that we developed in the previous chapter allow us to constrain roots of polynomials to any given location in the complex plane by inducing minimal perturbations in the coeffecients. Using the technique of *parametric minimization* in the spirit of Karmarkar and Lakshman [1996, 1997], we can take this idea one step further, namely to constraining one root of a given polynomial to an also given curve in the complex plane. The actual root locus is the one location (or an entire segment) on the curve that will result in minimal coefficient perturbations. An important application of this method to control theory will be presented in the next chapter.

First, we derive explicit formulae for the solution of the respective minimization problem in the case of complex coefficients. The proof presented here is more basic than the proofs in Karmarkar and Lakshman [1996, 1997]. In the next section, we derive algorithms for constraining a root to a given curve. We then take a brief look at the approximate GCD-problem and how to find nearest singular polynomials. Interestingly, while the case of complex coefficients leads easily to solutions in closed form, we were able to do this for real coefficients only in special cases so far. However, the situation seems to be reversed when we use the l^∞-norm in lieu of the Euclidean norm: we will derive an explicit formula for the minimum change in the infinity-norm for real coefficients and a real root in the third section.

3.1 Solutions for the l^2-Norm in Closed Form

For the case of complex coefficients and a single complex root α, i.e., for the minimization problem (2.1), we can derive explicit formulae for the minimum as functions in the parameter α. The expressions for the minimal coefficient perturbations are identical to the ones that were obtained by Karmarkar and Lakshman [1996, 1997] for approximate greatest common divisors, whereas the norm expression consists of the term for one polynomial only. For later use in parametric minimization, it is preferable to express the symbolic minimum in terms of the square of the norm of the coefficient perturbations in

order to avoid the square root.

We summarize the result in the following theorem:

Theorem 1 *For the symbolic least squares solution of (2.1), the square of the l^2-norm over the minimal perturbation is*

$$\mathcal{N}_{\min}(\alpha) = \|\delta\|^2 = \frac{\overline{f(\alpha)}f(\alpha)}{\sum_{k=0}^{n}(\overline{\alpha}\alpha)^k}\,, \tag{3.1}$$

while the perturbation of the j-th coefficient itself is

$$\delta_j = \frac{(\overline{\alpha})^j f(\alpha)}{\sum_{k=0}^{n}(\overline{\alpha}\alpha)^k}\,, \tag{3.2}$$

for $0 \leq j \leq n$, and assuming that $0^0 = 1$.

PROOF: The formulae can be obtained by either explicitly solving the normal equations, or equivalently, minimizing a quadratic form (see Karmarkar and Lakshman [1996]), or via a Lagrange multiplier approach (see Karmarkar and Lakshman [1997]). We give an alternate proof using basic vector geometry. It will not only improve our understanding of the subspaces that are involved, but it will also provide us with a more efficient method for the symbolic derivation of $\mathcal{N}_{\min}$ for problems that do not lead to solutions in closed form (see later in this chapter).

Because of lemma 1, the column vectors of $\mathbf{M}$ in (2.2) form a basis for an n-dimensional subspace of $\mathbb{C}^{n+1}$. Therefore, the unitary complement of the column-space is one-dimensional. It can easily be verified that

$$\mathbf{r} = [1, \overline{\alpha}, \overline{\alpha}^2, \ldots, \overline{\alpha}^n]^{tr}$$

is orthogonal to any column vector of $\mathbf{M}$, thus constituting a normal vector to the hyperplane spanned by the columns of $\mathbf{M}$.

The vector of the minimal perturbations δ is the orthogonal projection of $\mathbf{b} = [a_0, \ldots, a_{n-1}, a_n]^{tr}$ onto $\mathbf{r}$:

$$\delta = \frac{\mathbf{b}\cdot\mathbf{r}}{\mathbf{r}\cdot\mathbf{r}}\mathbf{r} = \frac{\mathbf{b}\cdot\mathbf{r}}{\|\mathbf{r}\|^2}\mathbf{r} = \frac{f(\alpha)}{\sum_{k=0}^{n}(\overline{\alpha}\alpha)^k}\mathbf{r}\,,$$

which is the vector $[\delta_0, \ldots, \delta_n]$ in (3.2). The norm expression becomes:

$$\mathcal{N}_{\min}(\alpha) = \|\delta\|^2 = \delta^*\delta = \frac{\overline{f(\alpha)}}{\|\mathbf{r}\|^2}\frac{f(\alpha)}{\|\mathbf{r}\|^2}\mathbf{r}^*\mathbf{r} = \frac{\overline{f(\alpha)}f(\alpha)}{\|\mathbf{r}\|^2},$$

which is (3.1). □

The following corollary states that $\mathcal{N}_{\min}$ is a positive semi-definite form in α:

Corollary 1 *The function $\mathcal{N}_{\min}(\alpha)$ is real-valued and positive semi-definite.*

PROOF: The numerator of $\mathcal{N}_{\min}(\alpha)$ is

$$\overline{f(\alpha)}f(\alpha) = \|f(\alpha)\|^2 \geq 0,$$

while for any value of α, its denominator $\|\mathbf{r}\|^2$ is strictly positive due to the 0th summand being $(\overline{\alpha}\alpha)^0 = 1$. By definition, both norm expressions are real-valued. □

Remarks

We note the following properties:

1. Lemma 1 guarantees that the least squares problem has a unique minimal solution. We saw in the previous chapter that this is not necessarily the case for other norms.

2. The denominator $\|\mathbf{r}\|^2$ of $\mathcal{N}_{\min}(\alpha)$ is also the determinant of the matrix $\mathbf{Q} = (\mathbf{M}^*)\mathbf{M}$. This matrix defines a quadratic form that was used in the original approach by Karmarkar and Lakshman [1996].

3. Every component of the perturbation vector δ contains $f(\alpha)$ as a factor. Therefore, δ is scaled by the leading coefficient a_n of f (or by any other coefficient for that matter). We can remove this degree of freedom by enforcing monicity (see problem 3). However, solutions of lower degree than n are not anymore achievable that way.

4. For consistency, it is important that for $j = 0$ and $\alpha = 0$, $(\overline{\alpha})^j = 1$. For that reason, the additional assumption for the otherwise mathematically undefined expression 0^0 was made in the statement of the theorem. With this addition, the formula is in accordance with lemma 2.

5. If we start out with a monic polynomial, and if we want to preserve monicity, we only have perturbations δ_j with $0 \leq j \leq n-1$, and the upper limit for the summation index in the denominator is also $n-1$. In our article for ISSAC'98, Hitz and Kaltofen [1998], we derived all formulae for the monic case.

3.2 Constraining a Root Locus to a Curve

We now want to use the formulae for the parametric minimum to develop an algorithm for computing the coefficients of a perturbed polynomial that has (at least) one root on a given curve $\Gamma \subset \mathbb{C}$. The basic idea is to substitute a *real* parametrization $\gamma(t)$ of Γ for the complex indeterminate α in $\mathcal{N}_{\min}(\alpha)$ of (3.1). The result will be an expression $\mathcal{N}_{\min}(\gamma(t))$ in the real parameter t. Local minima can be found at stationary points of this function, i.e., we will have to determine teh *real* solutions of the equation:

$$\frac{d\mathcal{N}_{\min}}{dt} = \frac{d\mathcal{N}_{\min}}{d\gamma}\frac{d\gamma}{dt} = 0\,. \tag{3.3}$$

For each solution τ_j, we obtain a root $\zeta_k = \gamma(\tau_j)$ of a perturbed polynomial that has at least one root $\zeta_j \in \Gamma$. We have to compare the respective values $\mathcal{N}_{\min}(\gamma(\tau_j))$ to find the global minimum. The validity of the method follows from corollary 1.

From this outline we see that the parametrization $\gamma(t)$ has to be differentiable. The parameter t may range over an interval $I \subset \mathbb{R}$. On the other hand, the curve Γ can be composed of finitely many segments, each with a parametrization $\gamma_k(t)$; the procedure has to be repeated for each segment in this case. We summarize our method in the following subsections, first for complex, and then for real coefficients. For both cases, let Γ be a piecewise smooth curve consisting of a finite number of segments, each having a parametrization

$$\gamma_k : I_k \to \mathbb{C}\,, \quad \text{where} \quad I_k \subset \mathbb{R}$$

is an interval that may cover all real numbers. We will identify the segment by its parametrization γ_k.

3.2.1 Complex Coefficients

For a given polynomial f with complex coefficients, the following algorithm describes how to compute the coefficients of a perturbed polynomial $\tilde{f}$ that has at least one root on Γ, while the perturbations are minimized in the l^2-norm. The parametric minimum $\mathcal{N}_{\min}(\gamma(t))$ of the norm expression after substitution will be abbreviated by $N(t)$. The jth zero of $N'(t)$ on segment γ_k of Γ is denoted by $\tau_{k,j}$.

Algorithm C:

Input: $f \in \mathbb{C}[z]$, and a curve Γ as specified above.

Output: $\tilde{f} \in \mathbb{C}[z]$, and $\tau \in \mathbb{R}$, s.t. $\tilde{f}(\gamma_k(\tau)) = 0$ for a segment γ_k of Γ, and $\|f - \tilde{f}\|_2 = \min$ over all segments.

C1: For each segment γ_k of Γ do:

- **C1.1:** Substitute $\gamma_k(t)$ for α in the symbolic minimum $\mathcal{N}_{\min}(\alpha)$ resulting in the expression $N(t)$.
- **C1.2:** Symbolically determine the derivative $N'(t)$.
- **C1.3:** Compute all *real* roots $\tau_{k,i}$ (of the numerator) of $N'(t)$. Select $\tau_{k,i}$ that minimizes $N(\tau_{k,i})$, and assign it to τ_k.

C2: From all $N(\tau_k)$ of step C1.3, determine the overall minimum $N(\tau)$.

C3: Compute the perturbations δ_j. Return $\tilde{f}$, k, and τ.

The complexity of the algorithm is depending on the parametrization. However, all steps of algorithm C finish in poynomial time. Symbolic simplifications during the substitution, and after computing the derivative can be costly, but computing the roots in C1.3 makes this the most expensive step. This is the place, where numerical techniques can be incorporated to gain speed over precision. By keeping corollary 1 in mind, bisection or quasi-Newton methods can be used in any case. If the parametrization is differentiable twice, we can apply full Newton iteration, yielding quadratic convergence.

Certain parametrizations have singular points which have to be treated separately,

like additional segments. E.g., the popular parametrization of the unit-circle:

$$\gamma(t) = \frac{t - \mathbf{i}}{t + \mathbf{i}},$$

for $-\infty < t < \infty$ leaves an infinitesimal gap at 1. In this case, we have to evaluate $\mathcal{N}_{\min}(1)$, and compare the result with the values obtained in C1.3. Another solution would be to use two half-circles whose parameter t only runs from -1 to $+1$.

Most of the time, Γ will consist of segments that have a parametrization in form of a rational function in t with no real poles. After substitution $N(t) = \mathcal{N}_{\min}(\gamma(t))$ will also be a rational function that has no real poles. Therefore, we only have to find the zeroes of the numerator of $N'(t)$, i.e., we have to solve a polynomial equation. We even could adapt our algorithm to only derive the numerator of the derivative by restricting the class of admissible parametrizations.

Using known bounds, we can restrict the search space for the real roots if we are given a precision limit that has to be achieved for the coefficient perturbations (see Karmarkar and Lakshman [1996, 1997] for the approximate GCD-problem).

Computer algebra systems are capable of computing the derivative for a wide range of parametrizations *symbolically*. However, numerical techniques (difference methods) could be used for cases where no explicit parametrization is available.

Evidently, the computations for individual segments γ_k can be performed in parallel. Because the inner loop (C1.1 to C1.3) contains all major computational steps, coarse-grain parallelization in a distributed environment is also feasible.

3.2.2 Real Coefficients

For polynomials with real coefficients, it appears to be more difficult to derive explicit formulae. We were able to obtain them in a few special cases, e.g., when Γ is a coordinate axis. For the important case of the imaginary axis (Hurwitz stability), we will present the formulae in the next chapter. For other cases, we can still compute symbolic solutions to the least squares problem, and use the result for substitution by a parametrization $\gamma(t)$. However, if we have to simplify huge expressions, this procedure can become rather costly. Trying to extend the set of explicit solutions will certainly be an important focus of future research activities.

For real coefficients and real perturbations, we have to distinguish two cases again: either the constrained root is real itself, or it belongs to a pair of conjugate complex values. Real roots can only occur if Γ has a non-empty intersection with the real axis. This set can consist of distinct intersection points or an entire segment γ_k, which would then be a real interval. Only part of a segment cannot lie on the real axis, as this would constitute a violation of the differentiability condition for γ_k. For the intersection points, we can directly evaluate formula (3.1) for $\mathcal{N}_{\min}$, whereas algorithm C can be used to find τ_k and $N(\tau_k)$ for a segment on the real axis.

In the second case (conjugate pair), we have to solve the minimization problem in (2.6) symbolically, and then substitute the expressions for the real and the imaginary part of $\gamma_k(t)$ into the derived solution accordingly. The following description of the algorithm encompasses both special cases:

Algorithm R:

Input: $f \in \mathbb{R}[x]$, and a curve Γ as specified above.

Output: $\tilde{f} \in \mathbb{R}[x]$, and $\tau \in \mathbb{R}$, s.t. $\tilde{f}(\gamma_k(\tau)) = 0$ for a segment γ_k of Γ, and $\|f - \tilde{f}\|_2 = \min$ over all segments.

R1: For the given polynomial $f \in \mathbb{R}[x]$, derive the symbolic least squares solution $\mathbf{w}_r(s,q)$ to (2.6). Expand the product $\mathcal{N}_{\min}(s,q) = (\mathbf{M}_r\mathbf{w}_r - \mathbf{b})^{tr}(\mathbf{M}_r\mathbf{w}_r - \mathbf{b})$.

R2: For each segment γ_k of Γ do:

R2.1: Substitute

$$\begin{aligned} s &\hookleftarrow -2\Re(\gamma(t)), \quad \text{and} \\ q &\hookleftarrow |\gamma(t)|^2 = \Re(\gamma(t))^2 + \Im(\gamma(t))^2 \end{aligned}$$

in the symbolic minimum $\mathcal{N}_{\min}(s,q)$, resulting in the expression $N(t)$.

R2.2: Symbolically determine the derivative $N'(t)$.

R2.3: Determine all *real* solutions $\tau_{k,i}$ to $N'(t) = 0$. Select $\tau_{k,i}$ that minimizes $N(\tau_{k,i})$, and assign it to τ_k.

R3: Use formula 3.1 to compute $\mathcal{N}_{\min}$ for the intersection points, and algorithm C from the previous section for common segments of Γ with the real axis.

R4: From the values for $\mathcal{N}_{\min}$ computed in all steps R2.3, as well as in step R3 select the minimum. Determine the perturbations using expressions from step R1, or formulae (3.2) respectively. Return $\tilde{f}$, k, and τ.

Most remarks from the previous section also apply to algorithm R. Here, all sub-steps of R2 and R3 can be executed in parallel. In fact, we could start step R3 already at the beginning of the algorithm, parallel to R1. This might be an advantage in terms of load-balancing, as step R2 will create several parallel processes, while R1 is typically executed on a single processor.

All sub-steps of algorithm R are known to run in polynomial time. Depending on the implementation of the least squares solver, the symbolic derivation in step R1 may take considerable time to complete. We present an alternate algorithm below. Instead of projecting the constant vector $\mathbf{b}$ (containing the coefficients of f) onto the range of $\mathbf{M}_r$, we can derive the perturbation vector δ directly by projecting $\mathbf{b}$ onto the orthogonal complement of $\mathbf{M}_r$, which is a two-dimensional subspace only. This method was also used in the proof of theorem 1, although for a complement of dimension one in that case. It will also be used to derive explicit formulae for Hurwitz polynomials with real coefficients in the next chapter. In the following algorithm P, we construct a projector matrix $\mathbf{P}$ from a orthonormal basis of the two-dimensional subspace that is the orthogonal complement of $\mathbf{M}_r$. The theoretical justification for this procedure can be found in section 2.4 of Golub and Van Loan [1983].

Algorithm P:

Input: $f \in \mathbb{R}[x]$, where $n = \deg(f)$.

Output: The vector of perturbations $\delta(s,q)$, as defined in (2.6), containing symbolic expressions in the indeterminates s and q.

P1: Construct the symbolic $(n+1)\times(n-1)$ matrix $\mathbf{M}_r$ in the indeterminates s and q.

P2: Derive basis vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ for $\ker(\mathbf{M}_r^{tr})$.

P3: Construct an orthonormal basis by executing a single Gram-Schmidt step:

$$\begin{aligned} \mathbf{u_1} &:= \frac{\mathbf{v_1}}{\|\mathbf{v_1}\|}, \qquad \mathbf{v} := \mathbf{v_2} - (\mathbf{v_2} \cdot \mathbf{u_1})\mathbf{u_1} \\ \mathbf{u_2} &:= \frac{\mathbf{v}}{\|\mathbf{v}\|} \end{aligned}$$

P4: assemble the $(n+1) \times 2$ matrix $\mathbf{V} = [\mathbf{u_1}|\mathbf{u_2}]$ from the column vectors $\mathbf{u_1}$ and $\mathbf{u_2}$.

P5: Compute the $(n+1) \times (n+1)$ subspace projector matrix $\mathbf{P} := \mathbf{V}\mathbf{V}^{tr}$.

P6: Project the coefficient vector of f by computing $\delta := \mathbf{Pb}$. Return $\delta(s,q)$.

Algorithm P makes use of standard functions that are part of the linear algebra package in all major computer algebra systems. The minimum norm expression that is used in algorithm R, can be computed in a single inner product:

$$\mathcal{N}_{\min}(s,q) = \delta(s,q)^{tr}\delta(s,q)\,.$$

3.3 Approximate Greatest Common Divisors and Nearest Singular Polynomials

In the previous section, we were given a one-dimensional variety for constraining the root locus. Solving the parametric minimization problem resulted in a real value τ. Now, if we look at two monic polynomials $f \in \mathbb{C}[z]$ and $g \in \mathbb{C}[z]$ with $\deg(f) = m$, $\deg(g) = n$, and $\mathrm{GCD}(f,g) = 1$, then we can use the symbolic solutions of theorem 1 to express the minimal perturbations that would transform f into a monic $\tilde{f}$, and g into a monic $\tilde{g}$, such that both $\tilde{f}$ and $\tilde{g}$ have a root α. In order for the perturbed polynomials to have a non-trivial GCD, they must have at least one common root. The condition for $\tilde{f}$ and $\tilde{g}$ to have α as a *common* root is reflected in the combined norm expression

$$\mathcal{N}_{\min}(\alpha) = \frac{\overline{f(\alpha)}\,f(\alpha)}{\sum_{k=0}^{m-1}(\overline{\alpha}\alpha)^k} + \frac{\overline{g(\alpha)}g(\alpha)}{\sum_{k=0}^{n-1}(\overline{\alpha}\alpha)^k} \tag{3.4}$$

that has to be minimized. If $\alpha = a + b\mathbf{i}$, where a and b are real indeterminates, the parametric minimization (see Karmarkar and Lakshman [1996, 1997]) proceeds by computing the real intersection points of the curves that are defined by $N_a = 0$ and $N_b = 0$, where

$$N_a = \text{numerator}\left(\frac{\partial \mathcal{N}_{\min}}{\partial a}\right), \quad \text{and} \quad N_b = \text{numerator}\left(\frac{\partial \mathcal{N}_{\min}}{\partial b}\right).$$

Whereas in the previous problems the root α was constrained to an explicitly defined subset of the complex or real numbers, here the set is given implicitly by the norm expression for the second polynomial. The fact that $\mathcal{N}_{\min}(\alpha)$ is just the sum of two expressions of theorem 1 is further evidence that out of the three questions in the introduction, Q1 is indeed the most fundamental one to ask, and that the methods developed in the previous sections can be used as building blocks for solving more advanced problems.

In the following, we want to illustrate how the related problem of finding a nearest singular polynomial can be expressed as minimization problem by using the same techniques as in chapter 2. Although one could state the problem for any given multiplicity, we restrict ourselves to a double root. The generalization is obvious, and is in the same spirit as problem 2.

Problem 5 *Given $f \in \mathbb{C}[z]$, monic, having only roots of multiplicity one. Find $\tilde{f} \in \mathbb{C}[z]$, monic, having a root $\alpha \in \mathbb{C}$ with multiplicity two (or higher), and $\|f - \tilde{f}\| = \min$.*

The perturbed polynomial $\tilde{f}$ will have a factor

$$(z-\alpha)^2 = z^2 - 2\alpha z + \alpha^2 .$$

We can again formulate the minimization problem in the language of linear algebra. We have to find

$$\|\delta\| = \min_{\mathbf{w} \in \mathbb{C}^{n-2}} \|\mathbf{M}_s \mathbf{w}_s - \mathbf{b}\| , \tag{3.5}$$

where

$$
\begin{aligned}
f(z) &= z^n + a_{n-1}z^{n-1} + \cdots + a_0\,, \quad n \geq 2\,, \\
\mathbf{b} &= [a_0, \ldots, a_{n-2} - \alpha^2, a_{n-1} + 2\alpha]^{tr} \in \mathbb{C}^n\,, \\
\mathbf{w}_s &= [u_0, \ldots, u_{n-3}]^{tr} \in \mathbb{C}^{n-2}\,,
\end{aligned}
$$

and

$$
\mathbf{M}_s = \begin{bmatrix}
\alpha^2 & & & & & \\
-2\alpha & \alpha^2 & & & \text{\Large 0} & \\
1 & -2\alpha & \alpha^2 & & & \\
 & \ddots & \ddots & \ddots & & \\
 & & 1 & -2\alpha & \alpha^2 & \\
 & \text{\Large 0} & & 1 & -2\alpha & \\
 & & & & & 1
\end{bmatrix} \in \mathbb{C}^{n\times(n-2)}\,. \tag{3.6}
$$

After computing a symbolic least squares solution for (3.5), we obtain expressions $\delta(\alpha)$ and $\mathcal{N}_{\min}(\alpha)$. Again, we have to find the stationary points of $\mathcal{N}_{\min}(\alpha)$, and select the overall minimum. The actual value of α will be computed during this process. If α is fixed, the problem becomes a special case of problem 2.

3.4 Parametric Solutions in the l^∞-Norm

In the previous chapter we reduced the problem of computing perturbations in the infinity-norm to a problem in linear programming. Deriving symbolic solutions in closed form seems to be difficult due to the fact that we would have to know the sign of an indeterminate in order to proceed locally in solving the linear program. At the end, we would be left with an exponential number of sign-pattern for the perturbation vector δ. As the result by Poljak and Rohn [1993] for nonsingularity of interval matrices suggests, we might be here very close to a NP-hard problem. However, we are able to actually derive

a symbolic solution for the infinity-norm of the minimal perturbation for the restricted problem of real coefficients and one real root.

Although we are able to find an expression for the minimum change in the norm, the method does not construct an actual polynomial $\tilde{f}$. Because we only consider real roots, the task of constraining a root to a curve Γ can be reduced to a basic linear programming problem of chapter 2, and therefore is not depending on an explicit solution. We only have to compute the intersection points of Γ with the real axis, and then select the one root location that causes the perturbation to be minimal. Obviously, we cannot apply this method to a segment γ_k that lies on the real axis, as this would result in a continuous set of solutions.

The derivation of the norm change in closed form is an application of the geometric method in Stiefel [1959], a method that is based on results by the Belgian mathematician de la Vallée-Poussin and goes back to the beginning of the century (see Vallée-Poussin [1911]). Our proof follows closely Stiefel [1959], although his intent is to find approximate solutions in the infinity ("Tschebyscheff") norm for generic over-determined linear systems of equations. Because of the special dimensions of our matrix **M**, we only have to perform one step in his deduction.
We state the result in the following theorem:

Theorem 2 *Let $f \in \mathbb{R}[x]$, and $\alpha \in \mathbb{R}$. Then there exists $\tilde{f} \in \mathbb{R}[x]$ such that $\tilde{f}(\alpha) = 0$, and $\|f - \tilde{f}\|_\infty = \min$. For any $\alpha \in \mathbb{R}$, the norm of the minimal perturbation is:*

$$\delta_\infty \stackrel{def}{=} \|\delta\|_\infty = \left| \frac{f(\alpha)}{\sum_{k=0}^{n} |\alpha^k|} \right| \tag{3.7}$$

PROOF: First we recall that the minimization problem can be stated as

$$\|\delta\| = \min_{\mathbf{w} \in \mathbb{R}^n} \|\mathbf{M}\mathbf{w} - \mathbf{b}\|, \tag{3.8}$$

where **b** contains the coefficients of f, and

$$\begin{aligned}
\mathbf{w} &= [u_0, \ldots, u_{n-1}]^{tr} \in \mathbb{R}^n \,, \\
\mathbf{M} &= \begin{bmatrix} -\alpha & & & & 0 \\ 1 & -\alpha & & & \\ & \ddots & \ddots & & \\ 0 & & 1 & -\alpha & \\ & & & 1 & \end{bmatrix} \in \mathbb{R}^{(n+1)\times n}
\end{aligned}$$

For the following, we assume that $\alpha \neq 0$. Because of lemma 2 we already know the solution for the special case $\alpha = 0$, and one can easily verify that the formula above is consistent with lemma 2. Without this assumption, the first row could become a zero vector, thus invalidating some of the arguments in our proof.

Each row in the linear system of (3.8) defines an equation of a hyperplane in $\mathbb{R}^n$:

$$M_{j,0}u_0 + M_{j,1}u_1 + \cdots + M_{j,n-1}u_{n-1} - a_j = 0 \,,$$

where $\mathbf{M} = \big(M_{j,k}\big)$, and $0 \leq j \leq n$. We know that **M** has rank n (lemma 1). If we would omit one equation, the remaining hyperplanes would intersect in a single point. All $n+1$ hyperplanes define a closed *simplex* in $\mathbb{R}^n$, and for arbitrary vectors **w**, we have *residues*

$$\delta_j = M_{j,0}u_0 + M_{j,1}u_1 + \cdots + M_{j,n-1}u_{n-1} - a_j \,.$$

Among the row vectors of **M**,

$$\mathbf{M}_{j,*} \stackrel{def}{=} \big[M_{j,0}, \ldots, M_{j,n-1}\big] \,,$$

there is a linear dependence that we already know from our proof of theorem 1:

$$\mathbf{M}^{tr}\mathbf{r} = 0 \,, \quad \text{thus} \quad \mathbf{r}^{tr}\mathbf{M} = 0 \,, \quad \text{where} \quad \mathbf{r} = [1, \alpha, \alpha^2, \ldots, \alpha^n] \tag{3.9}$$

(note that α is real, therefore $\overline{\alpha} = \alpha$).

Each row vector $\mathbf{M}_{j,*}$ is a *normal* vector to the corresponding hyperplane, and there-

fore to one of the facets that constitute the simplex. Stiefel [1959] is now able to characterize the inner points of the simplex. He calls such points "reference points". In our notation, the condition for a specific $\mathbf{w} \in \mathbb{R}^n$ to represent an inner point is that *either* the sign of the residue is

$$\begin{aligned} \text{for all} \quad 0 \le k \le n\,, \quad \operatorname{sgn} \delta_k &= \operatorname{sgn} \alpha^k\,, \quad \text{or} \\ \text{for all} \quad 0 \le k \le n\,, \quad \operatorname{sgn} \delta_k &= -\operatorname{sgn} \alpha^k\,. \end{aligned}$$

Now, because of (3.8)

$$\begin{aligned} \delta &= \mathbf{M}\mathbf{w} - \mathbf{b}\,, \quad \text{thereforeby}(3.9) \\ \mathbf{r}^{tr}\delta &= \mathbf{r}^{tr}\mathbf{M}\mathbf{w} - \mathbf{r}^{tr}\mathbf{b} = -\mathbf{r}^{tr}\mathbf{b}\,, \end{aligned}$$

or expanded

$$\sum_{k=0}^{n} \alpha^k \delta_k = -\sum_{k=0}^{n} \alpha^k a_k = -f(\alpha)\,. \tag{3.10}$$

If we incorporate the sign conditions from above then (3.10) becomes

$$\sum_{k=0}^{n} |\alpha^k|\,|\delta_k| = \pm \sum_{k=0}^{n} \alpha^k a_k\,. \tag{3.11}$$

Because of our assumption that $\alpha \neq 0$, we see that the absolute values $|\delta_k|$ of the residues are bounded for any $\mathbf{w}$ satisfying the sign rules above. Therefore, these rules characterize indeed inner points of the simplex.

Now, again following Stiefel [1959], we define the "center" of the reference points to be that point whose residues δ_k all have the same absolute value. I.e., for that center point, there exists $d \in \mathbb{R}$ such that

$$\delta_k = d \operatorname{sgn} \alpha^k\,, \quad \text{for} \quad 0 \le k \le n\,. \tag{3.12}$$

Combining (3.10) and (3.12) yields

$$d \sum_{k=0}^{n} \alpha^k \operatorname{sgn} \alpha^k = -f(\alpha)\,,$$

or

$$d = -\frac{f(\alpha)}{\sum_{k=0}^{n} |\alpha^k|}\,.$$

Finally, we have to define $\delta_\infty = |d|$ to arrive at the formula stated n the theorem. $\square$

We were able to obtain the symbolic minimum in one step because we have $n+1$ equations in n variables. In his paper, Stiefel uses the method iteratively to solve problems with m equations, where $m > n+1$, by selecting $n+1$ rows in each step, and moving from one "center" point to the next one. The algorithm resembles the simplex method.

Furthermore, one can use other theorems in his paper to show that δ_∞ is bounded from below by $\min_{0\le k\le n} |\delta_k|$, and from above by $\max_{0\le k\le n} |\delta_k|$, where $\delta = (\delta_k)$ is the *least squares* solution of problem (3.8).

In our derivation we make extensive use of sign rules. Therefore, there is little hope that a similar formula for the case of complex coefficients can be obtained via this approach.

CHAPTER 4
Computing the Radius of Stability

Recent results in robust stability can be categorized by two different ways of looking at the same problem: on the one hand, following the landmark paper of Kharitonov Kharitonov [1978a, 1979, 1978b], several *tests* for stability of a *given* family of polynomials have been devised (see Barmish and Kang [1993], Tempo [1989]). On the other hand, research focuses on the determination of the *radius of stability* for a given nominal polynomial and a certain norm in coefficient space. In this area, notably the work of Tsypkin and Polyak [1991], Desages et al. [1991], and Kogan [1992] provides necessary and sufficient conditions for the range of robust stability in a (weighted) l^p norm. However, these criteria allow actual *computation* of the stability radius only in a few special cases, as they leave infinite one- or two-dimensional sets to be searched for the minimum. Frequency domain plots seem to be the method of choice for more general applications (see Joya and Furuta [1991]).

Another approach is based on structured singular values, and its related theory; see for example Chen et al. [1994a,b]. Out of this area, Qiu and Davison [1989] comes closest to our method. While the first stage of their algorithm leads to expressions in closed form for a wide range of vector norms, the second stage still requires the use of some search method.

In the following sections, we present a method for computing the radius of stability in the l^2 norm for univariate polynomials with either complex or real coefficients. In terms of flavor, it is closest to the ideas presented in Pérez et al. [1994]. The method is applicable for a wide range of stability domains $\mathcal{D} \subset \mathbb{C}$. Among them are the familiar cases for Hurwitz and Schur stability. Applying the techniques from the previous chapter, we can reduce the problem to finding finitely many roots of an (algebraic) equation. Root-finding is a well-researched topic. The use of numerical methods in this stage leads to a hybrid symbolic-numeric algorithm, where the speed of numerical root-finders improves the overall performance.

For special domains, like the left half-plane, Kharitonov's theorem provides a way

of computing the radius of stability in the infinity norm. We could use binary search on the size of the intervals, to find a "corner" polynomial that would have a root on the domain boundary (i.e., the imaginary axis in the classical case). However, there might be other unstable polynomials in the family at that point, because Kharitonov's theorem provides us with a test for *stability* not instability. Still, the method is efficient, and gives us a radius of stability for the limited set of domains to which it is applicable.

Similar methods could be developed for l^1-like norms ("diamonds" of coefficients), as results like Tempo [1989] suggest.

After some preliminaries, we show how to use our method of constraining a root to a curve for computing the radius of stability in the l^2-norm, for complex and then real polynomials. For the special case of Hurwitz-stability, we are able to derive formulae in closed form also for polynomials with real coefficients.

4.1 Preliminaries

First, we give a definition for the stability of a generic domain of the complex plane: Let $\mathcal{D} \subset \mathbb{C}$ be an open, convex domain of the complex plane. The polynomial $f \in \mathbb{C}[z]$ (or $f \in \mathbb{R}[z]$) is called *$\mathcal{D}$-stable* if all its roots are located within $\mathcal{D}$.

For a polynomial family, we have to add the condition that all members have to be of the same degree in order to be $\mathcal{D}$-stable. This condition is especially important when we perturb a given stable polynomial until it becomes unstable. If the degree of the polynomial would be allowed to drop then it would "loose" one or several of its roots, and in most practical applications (e.g., stability of differential systems) this would change the structure of the problem.

Special cases are *Hurwitz* stability, if $\mathcal{D}$ is the open left half-plane, and *Schur* stability, if $\mathcal{D}$ is the open unit-disc around the origin. In the first case, $\mathcal{D}$ is unbounded, whereas in the latter case it is bounded.

Our goal is to derive a method for finding the nearest unstable polynomial $\tilde{f}$ for a given $\mathcal{D}$-stable polynomial f. Basis of our method is the following theorem, which could also be called the *inverse zero-exclusion principle*:

Theorem 3 *Let $f \in \mathbb{C}[z]$, be $\mathcal{D}$-stable ($\mathcal{D}$ as defined above), and let $\hat{f} \in \mathbb{C}$, be an unstable polynomial of the same or lower degree as f such that $\|f - \hat{f}\| = \varepsilon$, and $\varepsilon \in \mathbb{R}, \varepsilon > 0$.*

Then, there exists $\tilde{f} \in \mathbb{C}[z]$ *and* $\zeta \in \partial\mathcal{D}$ *such that* $\|f - \tilde{f}\| \leq \varepsilon$ *and* $\tilde{f}(\zeta) = 0$.

PROOF: For Hurwitz stability, see also Lemma 3.2 in Kogan [1992].

For $t \in \mathbb{R}$ and $0 < t \leq 1$, we define the polynomials f_t by

$$f_t(z) = f(z) + t \cdot (\hat{f}(z) - f(z)).$$

The roots of f_t move continuously as the parameter t varies over the interval $(0,1]$. For a proof of this property, see Marden [1966].

We want to discuss the case of a degree-drop first. If f has degree n, and $\deg(\hat{f}) = m$ with $m < n$, then as we increase t from zero to one, (at least) one of the roots of f_t will move to infinity. If the domain $\mathcal{D}$ is bounded, i.e., ∞ is not part of the contour $\partial\mathcal{D}$, then this root will have to cross $\partial\mathcal{D}$ somewhere, because of continuity, and we are done. Note that infinity cannot be in $\mathcal{D}$, otherwise it would not be an open set. Now, if $\mathcal{D}$ is unbounded (this is this for example the case for the left half-plane), the root can actually reach infinity. However, in this case, infinity belongs to $\partial\mathcal{D}$, and we are done as well.

If the degree does not drop along the line from f to $\hat{f}$ in coefficient space, then the expression

$$\|f - f_t\| = t \cdot \|\hat{f}(z) - f(z)\|$$

is a strictly increasing function in t for a generic vector norm. Because $\hat{f}$ is unstable, it must have one or more roots outside or on the boundary of $\mathcal{D}$. Again, by virtue of the continuous dependence of the roots on the coefficients, there must be $\tau \in (0,1]$ and $\zeta \in \partial\mathcal{D}$ such that $f_\tau(\zeta) = 0$ and $\|f - f_\tau\| \leq \varepsilon$. □

Because of theorem 3, the nearest unstable polynomial is one that has a root on $\partial\mathcal{D}$. Together with the methods presented in the previous chapter, we are now able to compute the radius of stability for domains $\mathcal{D}$ whose contour $\partial\mathcal{D}$ satisfies the input requirements for algorithms C and R. I.e., $\partial\mathcal{D}$ must have some suitable parametrization $\gamma(t)$, or must be composed of segments that have such parametrizations.

4.2 The Nearest Unstable Polynomial – Complex Coefficients

Given a domain $\mathcal{D} \subset \mathbb{C}$ and a $\mathcal{D}$-stable polynomial $f \in \mathbb{C}[z]$, we can basically use algorithm C to find $\tilde{f} \in \mathbb{C}[z]$ such that $\tilde{f}$ has a root on $\partial\mathcal{D}$. The first prerequisite is that $\partial\mathcal{D}$

is a piecewise smooth curve, having a parametrization $\gamma(t)$ in a real parameter t for each segment. If we allow the leading coefficient of f to be perturbed, and if $\mathcal{D}$ is unbounded, solutions that lead to a degree-drop of $\tilde{f}$ are valid, and we have to amend algorithm C to check that case. This additional test only has compute the absolute value of the leading coefficient and compare it against the norm of the minimal perturbations obtained in steps C1.3. It is evident that we achieve the minimum perturbation in the case of a degree-drop by setting the leading coefficient to zero, and leaving all other coefficients unperturbed.

If we want to preserve monicity, we have to use the modified formulae as mentioned in the remarks following theorem 1.

We illustrate the procedure in the following example:

Example 1 *The monic polynomial*

$$\begin{aligned} f(z) &= z^3 + (2.41 - 3.50\,\mathbf{i})z^2 + (2.76 - 5.84\,\mathbf{i})z \\ &\quad -1.02 - 9.25\,\mathbf{i} \end{aligned}$$

is Hurwitz, with approximate roots: $-1.04 + 3.10\,\mathbf{i}$, $-.99 - 1.30\,\mathbf{i}$, *and* $-.37 + 1.70\,\mathbf{i}$. *The boundary of the left half-plane can be parametrized by* $\gamma(t) = \mathbf{i}t$, *where* $t \in \mathbb{R}$. *We want to preserve monicity, therefore we will not have possible degree-drops.*

Consequently,

$$\begin{aligned} \mathcal{N}_m(\gamma(t)) &= (t^6 - 7.00t^5 + 12.5381t^4 + 9.6712t^3 \\ &\quad -18.1104t^2 - 62.9736t + 86.6029) \\ &\quad /(t^4 + t^2 + 1)\,. \end{aligned}$$

The numerator of the derivative is a monic polynomial of degree nine. It has three real roots (rounded to 6 digits): $\tau_1 = -1.84729$, $\tau_2 = -.248977$, *and* $\tau_3 = 1.88617$. $\mathcal{N}_m(\gamma(\tau_k))$ *evaluates to* 25.9376, 94.8227, *and* .284693 *respectively. Therefore,* τ_3 *leads to minimal perturbations. The* l^2*-norm distance (radius of stability) is the square root of* $\mathcal{N}_m$, *namely*

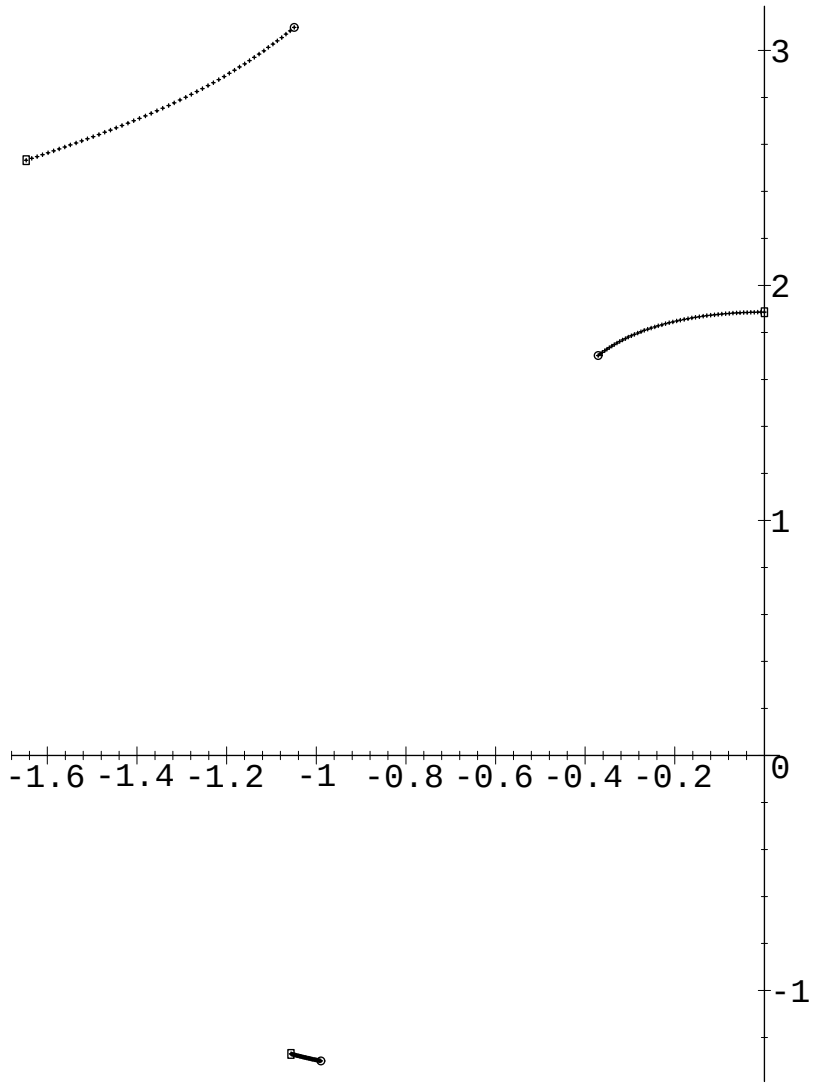

Figure 4.1: Hurwitz polynomial with complex coefficients

.533567. *Using (4), we can compute the coefficients of the perturbed polynomial* $\tilde{f}$*:*

$$\begin{aligned}\tilde{f}(z) &= z^3 + (2.7037 - 3.1492\,\mathbf{i})z^2 + (2.5740 - 5.6842\,\mathbf{i})z \\ &\quad -1.1026 - 9.3486\,\mathbf{i}\,.\end{aligned}$$

The roots moved to $-1.6472 + 2.5328\,\mathbf{i}$, $-1.0566 - 1.2698\,\mathbf{i}$, *and* $1.8862\,\mathbf{i}$ *respectively (see figure 1).*

4.3 The Nearest Unstable Polynomial – Real Coefficients

If $f \in \mathbb{R}[z]$, and we allow only real perturbations of the coefficients, we can use algorithm R to compute a nearest unstable polynomial. Again, if we allow degree-drops to happen then we have to add the additional test that we mentioned in the previous section. As already discussed in the outline of algorithm R, the case of real roots will actually be processed by algorithm C.

In the following small example for Schur-stability, we show the basic cases of algorithm R for a monic polynomial:

Example 2 *The polynomial* $f(z) = z^2 - 0.1z - 0.3$ *has (real) roots* -0.6 *and* 0.5*; both are inside the unit-circle. By perturbing the coefficients of* f*, one root can become either* -1 *or* $+1$*, or both can be a pair of complex conjugates located on the unit-circle. Using (3) from section 2, we have* $\mathcal{N}_{\min}(-1) = 0.32$ *and* $\mathcal{N}_{\min}(1) = 0.18$*. Alternatively, the perturbed polynomial* $\tilde{f}$ *would have roots* α *and* $\bar{\alpha}$*, where* $\alpha\bar{\alpha} = 1$*, i.e.,* $\tilde{f}(z) = (z-\alpha)(z-\bar{\alpha}) = z^2 - 2az + 1$*, and* $a = \Re(\alpha) \in \mathbb{R}$*. Perturbing the constant coefficient (*-0.3*) to a value of* $+1$ *would already give us a contribution of* $1.3^2 = 1.69$ *towards* $\mathcal{N}$*. Therefore, moving one root to* $+1$ *will result in the minimal perturbation.* $\tilde{f} = z^2 - 0.4z - 0.6$ *has another real root at* -0.6*.*

In the important case of *Hurwitz* stability, we can again derive explicit formulas. The special form of the domain boundary (imaginary axis) reduces (2.6) to a problem in a single parameter. In this case, $\tilde{f}$ will have a factor

$$(z - t\,\mathbf{i})(z + t\,\mathbf{i}) = z^2 + t^2\,, \quad \text{where} \quad t \in \mathbb{R}\,,$$

i.e., $s = 0$ and $q = t^2$. The matrix of the minimization problem for this special case is

$$\mathbf{M}_h = \begin{bmatrix} t^2 & & & & \\ 0 & t^2 & & \text{\Large 0} & \\ 1 & 0 & t^2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 0 & t^2 \\ \text{\Large 0} & & & 1 & 0 \\ & & & & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n-1)} . \tag{4.1}$$

We summarize the results in the following theorem:

Theorem 4 *Let $f \in \mathbb{R}[z]$ be Hurwitz. We define the polynomials $g, h \in \mathbb{R}[x]$, such that*

$$f(z) = g(z^2) + z \cdot h(z^2) ,$$

and substitute x for z^2. The nearest polynomial having at least one root on the imaginary axis at $z = t\,\mathbf{i}$, $t \neq 0$ is given by the following perturbations δ_j of the coefficients of f, where $0 \leq j \leq n$:

a) *If $n = \deg(f)$ is odd, and $m = (n-1)/2$, then*

$$\delta_j = \frac{\left(-t^2\right)^{j/2} g\left(-t^2\right)}{\sum_{k=0}^{m} t^{4k}} , \qquad j \text{ even,}$$

$$\delta_j = \frac{\left(-t^2\right)^{(j-1)/2} h\left(-t^2\right)}{\sum_{k=0}^{m} t^{4k}} , j \text{ odd,}$$

$$\mathcal{N}_{\min}(t) = \frac{g^2\left(-t^2\right) + h^2\left(-t^2\right)}{\sum_{k=0}^{m} t^{4k}} . \tag{4.2}$$

b) *If* $n = \deg(f)$ *is even, and* $m = n/2$*, then*

$$\delta_j = \frac{\left(-t^2\right)^{j/2} g\left(-t^2\right)}{\sum_{k=0}^{m} t^{4k}}, \qquad j \text{ even},$$

$$\delta_j = \frac{\left(-t^2\right)^{(j-1)/2} h\left(-t^2\right)}{\sum_{k=0}^{m-1} t^{4k}}, j \text{ odd},$$

$$\mathcal{N}_{\min}(t) = \frac{g^2\left(-t^2\right)}{\sum_{k=0}^{m} t^{4k}} + \frac{h^2\left(-t^2\right)}{\sum_{k=0}^{m-1} t^{4k}}. \tag{4.3}$$

PROOF: For Hurwitz polynomials, we could state the minimization problem as approximate GCD problem for the two polynomials g and h (this is also true for the case of complex coefficients, where $f(\mathbf{i}t) = g(t) + \mathbf{i}\,h(t)$ and $g, h \in \mathbb{R}[t]$), and use the techniques from Karmarkar and Lakshman [1996, 1997] to derive the minimum norm change and the perturbations. However, their methods also require both g and h to be monic, which condition is, in general, violated by one of the polynomials. Additionally, the derivation presented here, will give us more insights into the geometric properties of this minimization problem.

We are basically using the method of algorithm P to derive the minimal perturbations δ_j. For the real Hurwitz case, the column vectors of the matrix $\mathbf{M}_h$ in (4.1) form a basis for a $(n-1)$-dimensional subspace of $\mathbb{R}^{n+1}$. It can easily be verified that the two vectors

$$\mathbf{v}_1 = [1, 0, -t^2, 0, t^4, 0, \ldots], \quad \text{and} \quad \mathbf{v}_2 = [0, 1, 0, -t^2, 0, t^4, \ldots]$$

span the orthogonal complement of the column-space of $\mathbf{M}_h$. The vectors

$$\mathbf{u}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \quad \text{and} \quad \mathbf{u}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}$$

constitute an orthonormal basis for this 2-dimensional subspace. Therefore, the matrix

$$\mathbf{P} = \mathbf{V}\mathbf{V}^{tr}, \quad \text{where} \quad \mathbf{V} = [\mathbf{u}_1 \mid \mathbf{u}_2]$$

is a *projector* matrix in the sense of Golub and Van Loan [1983], sec. 2.4. As a linear map, it projects any vector from $\mathbb{R}^n$ onto the subspace spanned by $\mathbf{v}_1$ and $\mathbf{v}_2$. In our case,

we have to project the vector

$$\mathbf{b} = [a_0, \ldots, a_{n-1}, a_n],$$

i.e., we have to compute $\mathbf{Pb}$, in order to obtain δ. As one can see from the special structure of $\mathbf{M}_h$, $\mathbf{v}_1$ and $\mathbf{v}_2$, as well as the matrix $\mathbf{VV}^{tr}$, odd and even rows stay separated, which allows us to compute δ_j for odd and even numbers in terms of the coefficients of h and g, respectively. We also note, that odd rows in $\mathbf{VV}^{tr}$ are scaled by $1/\|\mathbf{v}_1\|^2$, even rows by $1/\|\mathbf{v}_2\|^2$, resulting in the denominators of the formulae (4.2) and (4.3). $\square$

Remarks:

1. In order to compute the actual minimum, one still has to execute steps R2.3, R3, and R4 of algorithm R. We could take the formal derivation even further, and explicitly write down the (polynomial) equations in t, that one has to solve. However, this can easily be done by a computer algebra system, and does not provide further insights into the problem.

2. Formulae (4.3) and (4.2) are consistent with the Hermite-Biehler theorem (Hermite [1879], Biehler [1879], Krein and Naimark [1981], Gantmacher [1959]). They represent the fact that g and h are perturbed independently, such that the perturbed polynomials $\tilde{g}$ and $\tilde{h}$ have $-t^2$ as a common root. The norm expressions $\mathcal{N}_{\min}$ are the ones we could derive from the formulae for the approximate GCD problem of two polynomials in the l^2-norm (see Karmarkar and Lakshman [1996, 1997]) by substitution.

In the case where $\mathcal{D}$ is the open unit-disc (*Schur* stability), the matrix $\mathbf{M}_r$ of (2.7) is also depending on a single parameter, namely s, as $q = 1$. Here however, the analogue of the *interlacing property* (as stated in the Hermite-Biehler theorem) does not apply to polynomials that are constructed from the coefficients of f directly (see Krein and Naimark [1981]). Therefore, the derivation of explicit formulae for $\mathcal{N}_{\min}(s)$ would not save us significant compute time over a direct solution of the least squares problem.

CHAPTER 5
Discussion

5.1 Conclusions

We demonstrated that minimal coefficient perturbations for constraining one or several roots of univariate polynomials can be computed by efficient algorithms. In particular, moving roots to prescribed locations in the complex plane leads to basic problems of linear algebra. We used the Euclidean norm, the infinity norm, and the l^1 norm of the coefficient vector to measure minimality of the perturbation.

For the Euclidean norm, we derived formulae for the minimal perturbations as a function of a single root locus in closed form. The derivation simplifies the result of Karmarkar and Lakshman [1996, 1997]. Based on this explicit representation are two polynomial-time algorithms for constraining a single root to a curve in the complex plane. We showed that the approximate GCD problem can be seen as a special case, where an explicit parametrization of the curve is replaced by the minimal norm expression for its coefficient perturbations of a second polynomial.

We were able to derive a formula in closed form that expresses the minimum of the perturbation in the infinity norm for the special case of real coefficients and a single real root.

The method of constraining a root locus to a curve was successfully used to compute the radius of stability in the Euclidean norm for a variety of stability domains. New formulae for the minimal perturbations, that make a real Hurwitz polynomial unstable, give further insights into the structure of the Hermite-Biehler criteria.

5.2 Future Directions

For practical implementations, the hybrid algorithms C and R, in particular, have to be fine-tuned for efficiency. It is desirable to have more explicit solutions to the least squares problem in the real case, as symbolic simplification can be costly. For special parametrizations of the curve segments, the solution space can be limited in function of a required output precision that has to be achieved.

Parametric minimization in the Euclidean norm has been proven to be successful, especially for complex coefficients. The concept has to be further investigated for other norms. The result for the special case of real coefficients and a single real root in the infinity norm exhibits an almost identical structure as its counterpart for the Euclidean norm. Comparing the minimum of $||\delta||$ for the infinity norm

$$||\delta||_\infty = \left| \frac{f(\alpha)}{\sum_{k=0}^{n} |\alpha^k|} \right|$$

with

$$||\delta||_2 = \sqrt{\mathcal{N}_{\min}(\alpha)} = \frac{|f(\alpha)|)}{\sqrt{\sum_{k=0}^{n} (|\alpha|^k)^2}}$$

in the Euclidean norm, leads to the question whether these expressions are special cases of a more general formula that would be valid for a generic norm.

We restricted ourselves to univariate polynomials here. Analogous questions can be asked for bivariate and multivariate polynomials. Of particular interest is the problem of finding *approximate factorizations* of bivariate polynomials. We mention the following basic problems concerning bivariate polynomials that seem to be within the scope of our methods for univariate polynomials:

Problem 6 *Given* $f \in \mathbb{C}[x,y]$ *and* $\alpha, \beta \in \mathbb{C}$*, find* $\tilde{f} \in \mathbb{C}[x,y]$ *of the same degree in x and y as* f*, such that* $\tilde{f}(\alpha,\beta) = 0$ *and* $||f - \tilde{f}|| = \min$.

Problem 7 *Given* $f \in \mathbb{C}[x,y]$ *and* $a,b,c \in \mathbb{C}$*, find* $\tilde{f} \in \mathbb{C}[x,y]$ *of the same degree in x and y as* f*, such that the linear term* $ax + by + c \mid \tilde{f}(x,y)$ *and* $||f - \tilde{f}|| = \min$.

Problem 8 *Given* $f \in \mathbb{C}[x,y]$ *such that* $f(\alpha,\beta) \neq 0$ *for all* $\alpha, \beta \in \mathbb{C}$*. Find* $\tilde{f} \in \mathbb{C}[x,y]$ *such that there exist* $\alpha, \beta \in \mathbb{C}$ *for which* $\tilde{f}(\alpha,\beta) = 0$ *and* $||f - \tilde{f}|| = \min$.

We can try to apply the concept of parametric minimization to other mathematical objects, especially to matrices. Although computing the distance to the nearest singular matrix is an NP-hard problem for a component-wise norm, this does not have to be be the case if the matrix has a special structure that has to be preserved under perturbation. The solution for the approximate GCD problem in the Euclidean norm automatically gives us

a polynomial-time algorithm for computing the nearest singular Sylvester matrix of the two polynomials in the Frobenius norm.

Finally, the techniques developed in chapter 2, can be applied to sensitivity analysis of ill-conditioned problems, like root finding.

BIBLIOGRAPHY

B. R. Barmish and H. I. Kang. A survey of extreme point results for robustness of control systems. *Automatica*, 29(1):13–35, 1993.

S. Barnett. Greatest common divisor of several polynomials. *Proc. Camb. Phil. Soc.*, 70: 269–268, 1971.

S. Barnett. *Polynomials and Linear Control Systems*. Marcel Dekker, New York, N.Y., 1983.

S. Barnett. *Matrices in Control Theory*. Robert E. Krieger, Malabar, F.L., 1984. rev. ed.

A. C. Bartlett, C. V. Hollot, and Huang Lin. Root locations of an entire polytope of polynomials: it suffices to check the edges. *Math. Control Signals Systems*, 1:61–71, 1988.

Biehler. Sur une classe d'équations algébraiques dont toutes les racines sont réelles. *J. Reine Angew. Math.*, 87:350–352, 1879.

D. L. Boley, T. J. Lee, and F. T. Luk. The Lanczos algorithm and Hankel matrix factorization. *Lin. Alg. Appl.*, 172:109–133, 1992.

Jie Chen, K. H. Fan, and C. N. Nett. Structured singular values and stability analysis of uncertain polynomials, part 1: the generalized μ. *Systems & Control Letters*, 23(1): 53–65, 1994a.

Jie Chen, K. H. Fan, and C. N. Nett. Structured singular values and stability analysis of uncertain polynomials, part 1: a missing link. *Systems & Control Letters*, 23(2): 97–109, 1994b.

T. F. Coleman and Yuying Li. A global and quadratically convergent method for linear l^∞ problems. *SIAM J. Numer. Anal.*, 29(4):1166–1186, 1992a.

T. F. Coleman and Yuying Li. A globally and quadratically convergent affine scaling method for linear l_1 problems. *Mathematical Programming*, 56:189–222, 1992b.

R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. In *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation, ISSAC'95*, pages 195–207. ACM Press, 1995.

S. Dasgupta. Kharitonov's theorem revisited. *Systems and Control Letters*, 11:381–384, 1988.

J. Demmel. Steve Smale and the geometry of ill-conditioning. Technical report, Mathematics Department and Computer Science Division, University of California, Berkeley, CA, 1991.

J. W. Demmel. On condition numbers and the distance to the nearest ill-posed problem. *Numerische Mathematik*, 51:251–289, 1987.

J. W. Demmel. The componentwise distance to the nearest singular matrix. *SIAM J. Matrix Anal. Appl.*, 13:10–19, 1992.

A. C. Desages, L. Castro, and Cendra H. Distance of a complex coefficient stable polynomial from the boundary of the stability set. *Multidimensional Systems and Signal Processing*, 2:189–210, 1991.

J. Doyle. Analysis of feedback systems with structured uncertainty. *IEE Proc. Part D*, 129:242–250, 1982.

Joseph G. Ecker and Michael Kupferschmid. *Introduction to Operations Research*. John Wiley & Sons, New York, N.Y., 1988.

I. Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate polynomial GCDs. *Journal of Pure and Applied Algebra*, 117/118:229–251, 1997.

I. Z. Emiris and V. Y. Pan. The structure of sparse resultant matrices. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation, ISSAC'97*, pages 189–196. ACM Press, 1997.

Y. K. Foo and Y. C. Soh. Root clustering of interval polynomials in the left-sector. *Systems & Control Letters*, 13:239–245, 1989.

R. A. Frazer and W. J. Duncan. On the criteria for stability for small motions. *Proceedings of the Royal Society A*, 124:642–654, 1929.

F. R. Gantmacher. *Matrix Theory*. Chelsea Publishing Company, New York, N.Y., 1959.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, M.D., 1st edition, 1983.

C. Hermite. Sur l'indice des fractions rationelles. *Bulletin Soc. Math. France*, 7:128–131, 1879.

M. A. Hitz and E. Kaltofen. Efficient algorithms for computing the nearest polynomial with constrained roots. accepted for publication in Proc. ISSAC'98, 1998.

D. G. Hough. *Explaining and ameliorating the ill condition of zeros of polynomials*. PhD thesis, University of California, Berkeley, C.A., 1977.

V. Hribernig and H. J. Stetter. Detection and validation of clusters of polynomial zeros. *JSC*, 24(6):667–681, 1997.

A. Hurwitz. Über die Bedingungen unter welchen eine Gleichung nur Wurzeln mit negativen reellen Teilen besitzt. *Math. Annalen*, 46:273–284, 1895.

K. Joya and K. Furuta. Criteria for Schur and $\mathcal{D}$-stability of a family of polynomials constrained with l^p-norm. In *Proc. IEEE Conf. Dec. and Control*, pages 433–434, 1991.

W. Kahan. Numerical linear algebra. *Can. Math. Bull.*, 9:757–801, 1966.

E. Kaltofen. Computing the irreducible real factors and components of an algebraic curve. *AAECC*, 1:135–148, 1990.

E. Kaltofen. Polynomial factorization 1987-1991. In I. Simon, editor, *Proc. LATIN '92*, number 583 in Springer Lect. Notes Comput. Sci., pages 294–313, 1992.

N. Karcanias and M. Mitrouli. A matrix pencil based numerical method for the computation of the GCD of polynomials. In *Proc. IEEE Conf. Decision and Control*, pages 425–430, 1992.

N. Karcanias and M. Mitrouli. A matrix pencil based method for the computation of the GCD of polynomials. *IEEE Trans. Auto. Control*, 39(5):977–981, 1994.

N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

N. Karmarkar and Y. N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. In *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC'96*, pages 35–39, 1996.

N. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *Submitted for publication*, 1997.

V. L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differensial'nye Uravnenya*, 14(11):2086–2088, 1978a.

V. L. Kharitonov. On a generalization of a stability criterion. *Akademii nauk Kazakhskoi SSR, Seria fiziko-mathematicheskaia*, 1:53–57, 1978b.

V. L. Kharitonov. Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differential Equations*, 14:1483–1485, 1979.

Donald E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, Reading, M.A., 3 edition, 1997.

J. Kogan. How near is a stable polynomial to an unstable polynomial? *IEEE Trans. Circuits Syst. I*, CAS-39(8):676–680, 1992.

M. G. Krein and M. A. Naimark. The method of symmetric and Hermitian forms in the theory of the separation of the roots of algebraic equations. *Linear and Multilinear Algebra*, 10:265–308, 1981. Originally published in Russian, Kharkov (1936).

D. Lazard. Résolution des systèmes d'équations algébriques. *Theoretical Computer Science*, 15:77–110, 1981.

Morris Marden. *Geometry of Polynomials*. Mathematical Surveys, *Number 3*. American Mathematical Society, Providence, R.I., 2nd edition, 1966.

R. J. Minnichelli, J. J. Anagnost, and C. A. Desoer. An elementary proof of Kharitonov's stability theorem with extensions. *IEEE Trans. Automatic Control*, 34(9):995–998, 1989.

Matu-Tarow Noda and Tateaki Sasaki. Approximate GCD and its application to ill-conditioned algebraic equations. *Journal of Computational and Applied Mathematics*, 38:335–351, 1991.

V. Pan. Complexity of computations with matrices and polynomials. *SIAM Review*, 34: 225–262, 1992.

V. Y. Pan. Approximate polynomial GCDs, padé approximation, polynomial zeros, and bipartite graphs. accepted for publication in Proc. SODA'98, 1998.

F. Pérez, C. Abdallah, and Docampo D. Robustness analysis of polynomials with linearly correlated uncertain coefficients in l^p-normed balls. In *Proc. IEEE Conf. Dec. and Control*, pages 2996–2997, 1994.

S. Poljak and J. Rohn. Checking robust nonsingularity is NP-hard. *Math. Control, Signals, and Systems*, 6:1–9, 1993.

L. Qiu and E. J. Davison. A simple procedure for the exact stability robustness computation of polynomials with affine coefficient perturbations. *Systems and Control Letters*, 13:413–420, 1989.

A. Rantzer. Stability conditions for polytopes of polynomials. *IEEE Trans. Auto. Control*, 37(1):79–89, 1992.

J. Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40:59–93, 1988.

J. Renegar. Incorporating condition measures into the complexity theory of linear programming. Technical report, Cornell University, Ithaca, New York, 1992a.

J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: part I: introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *J. Symbolic Comput.*, 13(3): 255–299, 1992b.

J. Rohn. Systems of linear interval equations. *Lin. Alg. Appl.*, 126:39–78, 1989.

A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Tech. report, Univ. Tübingen, 1982.

A. Schönhage. Quasi-GCD computations. *Journal of Complexity*, 1:118–137, 1985.

H. R. Schwarz. *Numerische Mathematik.* B. G. Teubner, Stuttgart, 1986.

K.-P. Sondergeld. A generalization of the Routh-Hurwitz stability criteria and an application to a problem in robust controller design. *IEEE Trans. Auto. Contr.*, AC-28(10): 965–970, 1983.

E. Stiefel. Über diskrete und lineare Tschebyscheff-Approximationen. *Numerische Mathematik*, 1:1–28, 1959.

E. Stiefel. Note on Jordan elimination, linear programming and Tschebyscheff approximation. *Numerische Mathematik*, 2:1–17, 1960.

R. Tempo. A simple test for Schur stability of a diamond of complex polynomials. In *Proc. IEEE Conf. Decision and Control*, pages 1892–1895, 1989.

L. N. Trefethen and D. Bau III. *Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

Y. Z. Tsypkin and B. T. Polyak. Frequency domain criteria for l^p-robust stability of continuous linear systems. *IEEE Trans. Auto. Control*, AC-36:1464–1469, 1991.

E. E. Tyrtyshnikov. How bad are Hankel matrices? *Numerische Mathematik*, 67:261–269, 1994.

Ch. J. de la Vallée-Poussin. Sur la méthode de l'approximation minimum. *Soc. Scient. Bruxelles, Annales*, 2ième *partie, mémoires*, 35:1–16, 1911.

J. R. Vera. Ill-posedness and the complexity of deciding existence of solutions to linear programs. Technical report, Cornell University, Ithaca, New York, 1992a.

J. R. Vera. Ill-posedness and the computation of solutions to linear programs with approximate data. Technical report, Cornell University, Ithaca, New York, 1992b.

J. H. Wilkinson. *Rounding Errors in Algebraic Processes.* Prentice-Hall, Englewood Cliffs, N.J., 1963.

J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, England, 1965.

J. H. Wilkinson. The perfidious polynomial. In Gene H. Golub, editor, *Studies in Numerical Analysis*, volume 24 of *Studies in Mathematics*, pages 1–28. M.A.A., 1984.

D. M. Young and R. T. Gregory. *A Survey of Numerical Mathematics*, volume 2. Dover Publications, Mineola, N.Y., 1973.

P. M. Young and J. C. Doyle. Computation of μ with real and complex uncertainties. In *Proc. IEEE Conf. Decision and Control*, pages 1230–1235, 1990.

APPENDIX A
Approximate Solutions to Inconsistent Systems

Here, we describe how to compute approximate solutions to overdetermined linear systems of real equations

$$\sum_{j=1}^{n} a_{k,j} x_j = b_k \,, \quad \text{for} \quad 1 \le k \le m \,, \quad \text{where} \quad m \ge n \,,$$

in the l^∞ and l^1-sense, by re-formulating the problem as a linear program. We follow Ecker and Kupferschmid [1988], section 2.3; an early reference for the l^∞-case is Stiefel [1960].

We include simple implementations in MAPLE that were not intended to be efficient. However, as a utility for checking the validity of concepts, they are a valuable tool.

A.1 Solutions in the l^∞-Norm

An approximate l^∞-solution $\mathbf{x} \in \mathbb{R}^n$ has to minimize

$$\max_{1 \le k \le m} |\delta_k| \,, \quad \text{where} \quad \delta_k = \sum_{j=1}^{n} a_{k,j} x_j - b_k \,.$$

We introduce an additional free variable w, and formulate the problem as a linear program with $2m$ constraints:

$$\begin{array}{lll} \text{Minimize:} & w & \\ \text{Subject to:} & w \ge \quad \delta_k & 1 \le k \le m \\ & w \ge -\delta_k & 1 \le k \le m \end{array} \tag{A.1}$$

The procedure `infsolve` is a MAPLE implementation that uses the built-in SIMPLEX package to find solutions to the linear program (A.1). For brevity, additional validity checks in the program code were left out.

```
## Input: real mxn matrix A, where m >= n,
##        and a m-dimensional vector b of real constants.
##
## Output: n-dimensional vector x, such that ||Ax - b|| = min
##         where ||.|| is the infinity-norm
infsolve := proc (A,b)

  local m, n, i, j, c, d, w, x, s, ss;

  m := linalg[rowdim](A);
  n := linalg[coldim](A);
# set up local vectors
  x := linalg[vector](n);
  d := linalg[vector](m);
  c := {};   # set of constraints
# generate linear equations, and constraints
  for i from 1 to m do
    d[i] := -b[i];
    for j from 1 to n do
      d[i] := d[i] + A[i,j]*x[j];
    od;
    c := c union { w>=d[i], w>=-d[i] };
  od;
# solve linear program
  ss := simplex[minimize](w,c);
# assign solution to output vector
  for i from 1 to n+1 do
    s := op (i,ss);
    if op (1,s) = 'w' then
      print (s);
    else
```

```
      assign (s);
    fi;
  od;

  RETURN (evalm(x));
end;
```

A.2 Solutions in the l^1-Norm

In the l^1-norm, we have to find $\mathbf{x} \in \mathbb{R}^n$ that minimizes

$$\sum_{k=1}^{m} |\delta_k| \,, \quad \text{where} \quad \delta_k = \sum_{j=1}^{n} a_{k,j} x_j - b_k \,.$$

With $2m$ additional variables d_k^+ and d_k^-, as well as $3m$ constraints, we obtain the linear program:

$$\begin{array}{lll} \text{Minimize:} & \sum_{k=1}^{m} \left(d_k^+ + d_k^-\right) & \\ \text{Subject to:} & d_k^+ - d_k^- = \delta_k & 1 \le k \le m \\ & d_k^+ \ge 0 & 1 \le k \le m \\ & d_k^- \ge 0 & 1 \le k \le m \end{array} \tag{A.2}$$

As noted in Ecker and Kupferschmid [1988], either d_k^+ or d_k^- (or both) will be equal to zero for each k. The minimal solution to the linear program (A.2) will automatically satisfy that condition.

The MAPLE procedure `onesolve` also takes a matrix and a vector of constants as input, and returns the solution vector.

```
onesolve := proc (A,b)

  local m, n, i, j, c, d, dm, dp, w, x, s, ss;

  m := linalg[rowdim](A);
```

```
  n := linalg[coldim](A);
# set up local vectors
  x := linalg[vector](n);
  d := linalg[vector](m);
  dm := linalg[vector](m);
  dp := linalg[vector](m);

  c := {};   # set of constraints
  w := 0;    # objective function
# generate linear equations, and constraints
  for i from 1 to m do
    d[i] := -b[i];
    for j from 1 to n do
      d[i] := d[i] + A[i,j]*x[j];
    od;
    w := w + dm[i] + dp[i];
    c := c union { dp[i]-dm[i]=d[i], dm[i]>=0, dp[i]>=0 };
  od;
# solve linear program
  ss := simplex[minimize](w,c);
# assign solution to output vector
  for i from 1 to nops(ss) do
    s := op (i,ss);
    assign (s);
  od;

  RETURN (evalm(x));
end;
```