University of California

Santa Barbara

Roots and Critical Points of Complex Polynomials: Applications of Algorithms in Real Algebra, Moment Theory, Convex Analysis, Optimization, and Positive Polynomials to a Conjecture in Pure Mathematics

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy

in

Mathematics

by

Richard William Spjut

Committee in charge:

Professor Mihai Putinar, Chair

Professor Bassam Bamieh

Professor Jeffrey Stopple

June 2020

The dissertation of Richard William Spjut is approved.

Professor Bassam Bamieh

Professor Jeffrey Stopple

Professor Mihai Putinar, Chair

June 2020

Roots and Critical Points of Complex Polynomials:

Applications of Algorithms in Real Algebra, Moment Theory, Convex Analysis, Optimization, and Positive Polynomials to a Conjecture in Pure Mathematics

by

Richard William Spjut

Acknowledgements

There are so many people to thank and many thanks! So many people that we cannot mention them all here for our own finiteness, finite resources, and humble human bounds on throughput. If you find an organization or event in which we attended, for which you worked, you supported, or for which you worked or supported as a part of a supporting organization to those organizations (and so on), then please consider mention of it in our C.V. (or here in the acknowledgements) as a sincere expression of gratitude. Similarly for any people whose work is referenced in our Bibliography. It is indeed, a debt of gratitude.

Our less-mathematically-inclined friends, family, and supporters might enjoy references [81] and [19]. I am grateful for the many such people who have raised, befriended, guided, or supported me in my life!! Thank you!

[81] is accessible to laypeople; it includes instructions about how to read it as someone who does not identify as a mathematician. At some point during graduate school, Mihai gave me a copy of it that I consider a treasured gift. It has also served as a resource for putting much of the mathematics I studied while in graduate school into historical context, especially work that follows (or is related to) Emil Artin's work on Hilbert's Seventeenth Problem. In fact, even as we read through it again now, the following sentence, "Artin commented that he preferred a clear existence proof to a construction with $2^{2^{100}}$ steps" bears deep connections to my work. Indeed, computational complexity of algorithms for some general decision problems in Real Algebra are doubly exponential. This is a crux of powerful languages. Loosely speaking: it's a trend that the more powerfully expressive a consistent, formal language is – the more humble human knowledge within that language is [especially with respect to completeness and complexity]. I bear witness to the awesome power of our Good LORD that we can be thusly humbled. And, I'm grateful when we're graced with clear, concise, constructive proofs.

[19] contains a section regarding "computationally kindness" to which I aspire. The "computationally kind" thing to do, in my mind, is to explain clearly some shortcomings of my work: this is a humble thesis. It does not do justice to depths of mathematics. We have yet to resolve Conjecture 0.1 and some of the methods I describe herein are still doubly-exponential in complexity. Note that as soon as we obtain any certificate similar to 3.5 then the complexity is drastically reduced. Indeed, I'll state with risk of seeming asinine: our complexity for checking known certificates is constant. Alas, I did not obtain a number of results (analogs of 3.5 for larger n), known to exist (for $3 \leq n \leq 8$), that I was hoping to obtain prior to submitting this document. Regardless, I made a commitment to Mihai, Medina, friends, family and coworkers to submit this document by May, 2020.

To any of my nonmathematican readers or students interested in mathematics who have yet to study much of the other references in this dissertation, I wish to provide to you the encouragement of Paul Halmos, "*The beginner... should not be discouraged if... [one] finds that [one] does not have the prerequisites for reading the prerequisites.*" [63]. Regarding someone interested in graduate school in Mathematics or a graduate program in Mathematics, one reference which expanded my mind toward other choices of agency is [35].

Words cannot express how patient and kind Mihai, our doctoral advisor, was with me. Moreover, I'm grateful for all of the courses, references, and communication with him along the way. I know I have not been as prolific as other people in mathematics communities. I take responsibility for it. Mihai, amongst numerous others, encouraged me to publish early, often, and much sooner and much more than I have.

Similarly, I'm grateful for the patience and work of Medina Price, our Graduate Student Advisor. Thank you for your kindness.

Special thanks to

- at Caltech (2000-2004): Prof. Dr. Barbara Green (Dean of Undergraduate Students), Prof. Dr. Nikolai Makarov, Prof. Dr. Wilhelm Schlag, Prof. Dr. Daniel

Groves, Prof. Dr. Michael Goldberg, and Dr. Bryan Smith (Dabney Hovse R.A.).

- at Humboldt State University (HSU) and Eureka High School (1996-2000): Prof. Dr. Jeff Haag, Prof. Dr. Sharon Brown, Mr. Weldon Benzinger (Physics and Chemistry teacher and organizer of Science Bowl team), Mr. Roger Smith (trigonometry and calculus teacher), and Doctora Patricia Maisch (statistics teacher).

Indeed, I must also recall the cautionary words Professor Jeff Haag. Circa 1998-2000 there was a math seminar at HSU given by a visitor on a subject in the field of mathematics which involved complex numbers. Jeff basically said, "beware of statements in Complex Analysis which we've yet to prove." There is an intellectual depth worthy of our respect in which mathematical themes such as axiomatic independence relate to resolution of conjectures. So, we tread lightly.

Thank you to all my teachers, school administrators and supporters during my formative years, especially including: Makakilo Baptist Pre-School teachers on Oahu, Mrs. Miller, Mrs. Gamboa (Blanc), Mr. Bill Hawkins, and Mrs. Lamont. These people deserve a lot of credit for providing a constancy as exemplary role models of dignity and for teaching with firm, gentle energy.

"HONOUR *thy father and they mother: that thy days may be long upon the land which the* LORD *thy God giveth thee*" — Exodus 20:12

Thanks to my mom and dad for loving and supporting me throughout my development: your presence, time, shelter, energy, food, clothing, reading to me, and care. Much love and aloha to my family and our family friends, past, present, and future. Especially my grandmas Frances and Mary Bo. God Bless you and keep you.

"Use was made of the computational facilities administered by the Center for Scientific Computing at the CNSI and MRL (an NSF MRSEC; DMR-1720256) and purchased through NSF CNS-1725797." Thank you to Paul Weakliem and Nathan Rogers of CNSI.

We made use of research gained from NSF grants DMS-0701094, DMS-0350911, DMS-0923839, DMS-1001071 and many other works supported by the National Science Foun-

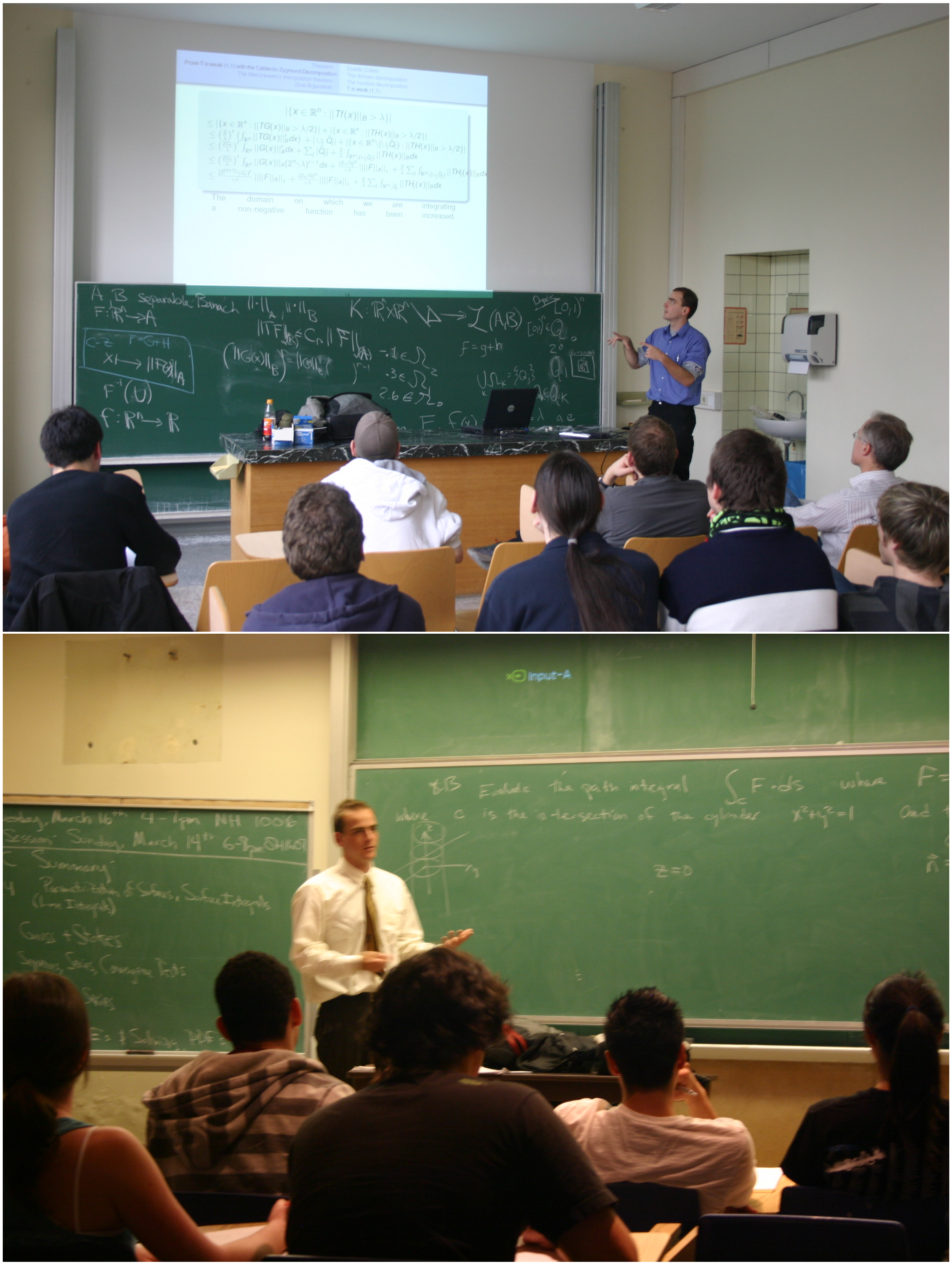


Figure 1: For the sake of my family and my teachers, we share two photos: teaching on March 12^{th}, 2010 and presenting on November 20^{th}, 2009. These pictures exist because Grandma Frances asked me once to please take a picture of me teaching and for kindness of our audiences.

dation.

Thank you so much to the many people of the following communities: Free Software Foundation, GNU, Debian, Ubuntu, Apache, LaTeX, BibTeX, arXiv, American Mathematical Society (AMS), Mathematical Association of America (MAA), Mathematics Geneaology Project, Society for Industrial and Applied Mathematics (SIAM), Mathworks and Matlab, Maple, Magma, Macaulay 2, Wolfram, Octave, Singularity, Singular, SageMath, Oracle VirtualBox, Sun Microsystems and Oracle Java Development Kit, Apache Software Foundation, Commons Math, Maven, LAPACK, Borland C Compilers, git, Borland Starteam, Scala, Python, bash,grep, nano, vim, sed, awk, Notepad++, Netbeans, JetBrains, SonarQube, Jupyter, Zeppelin, Git, Github, ePaul's ltxdoclet, YALMIP, GLOPTIPOLY, SOSTOOLS, SDPA, STPT3, SparsePOP, Mosek, Texas Instruments, Basic, DOS, Microsoft Windows, Apple Mac OSX, Intel, UC Berkeley and Mathematical Science Research Institute (MSRI), UC Los Angeles (UCLA) and Institute for Pure and Applied Mathematics (IPAM), UC Davis and Galois Group, UC San Diego (UCSD), Uni Bonn, Toshiba, Dell, and several independent computer parts manufacturers, all publishers of mathematics books, articles, documents. The many universities and mathematical institutions throughout the world. The people of every tribe and nation which has supported, and continues to support, mathematics research. I would also like to thank veterans of the United States of America and our allies: past, present and future. I lack the words to express this depth of gratitude to you.

To my "mathematical brother," Martin Harrison: thank you for your responses to "poorman published" articles I sent you, especially circa 2018-2020. Basically, to echo the acknowledgements in your dissertation: thank you for doing mathematics with me. To Dev Kumar: thank you for your support, especially our Quarterly Reporting system circa 2019-2020. To our proofreaders, including Brittany Erickson, Travis Waddington, Jón Karl Sigurðsson: thank you for being open to proofreading. The usual disclaimer applies: any remaining errors, including omissions, are my sole responsibility.

Curiculum Vitæ

Richard William Spjut

Education

Degree	Date	Institution	Advisor	Subject
Ph.D.	June 2020	UCSB	Mihai Putinar	Mathematics
Defense	May 2012	UCSB	Mihai Putinar	Mathematics
M.Sc.	2008	UCSB	Mihai Putinar	Mathematics
B.Sc.	2004	Caltech	Nikolai Makarov	Mathematics

Seminars, Conferences and Talks

Date	Venue
2019	Geometry of Polynomials at the Simons Institute for the Theory of Computing of UCB
2018	Robust and High-Dimensional Statistics at the Simons Institute for the Theory of Computing of UCB
2015	Humboldt State University Colloquium
2014	Humboldt State University Colloquium
2011	Lecture Series on Convex Optimization by Stephen Boyd at UCSB [Peter Kokotovic]
2007-2011	Analysis Seminar at UCSB Math Dept. Co-Organizer
2010	Convex Optimizsation and Algebraic Geometry at IPAM of UCLA
2010	Mutlivariate Operator Theory at BIRS in Banff, Canada
2010	Algebraic, Geometric, and Combinatorial methods for Optimization at MSRI of UCB
2010	Gene Golub Graduate Summer School in Bari, Italy
2010	Linear Matrix Inequalities and Polynomial Optimization (LMIPO) at UCSD
2009	PDEs Seminar at Bonn International Graduate School (BIGS), Bonn, Germany
2009	Scientific Computing Seminar at BIGS, Bonn, Germany
2009	CMFT at Bilkent University, Anakara, Turkey
2009	INRIA at Sophia Antipolis, France
2009	South East Analysis Meeting (SEAM) at University of South Florida, St. Petersburg, Florida
2008	Hilbert Spaces of Analytics Functions at CRM of the University of Montréal, Canada
2008	Multivariate Operator Theory at TAMU Math Dept, College Station, Texas
2008	American Mathematical Society (AMS) Spring Western Section Meeting at Pomona College, Cafifornia
	Special Session on Operators, Functions and Linear Spaces, III
2007	Continuous Optimization and Applications at MSRI
2006-2007	Graduate Student Seminar at UCSB Math Dept. Organizer
2006-2007	Operator Algebras Seminar at UCSB Math Dept.
2006	Hypatian Seminar at UCSB Math Dept.
2005	Colloquium Lecture at HSU Math Dept.

Abstract

Roots and Critical Points of Complex Polynomials:

Applications of Algorithms in Real Algebra, Moment Theory, Convex Analysis, Optimization, and Positive Polynomials to a Conjecture in Pure Mathematics

by

Richard William Spjut

Conjecture 0.1 (Conjecture of Blagovest Sendov (1958))
For a complex polynomial of degree two or more with all its roots contained within the closed unit disk, each root has a critical point within unit distance.

We introduce a countable collection of conjectures – one for each degree – by transferring into languages of Real Algebra. For fixed degree, each conjecture is decidable. Thus, we consider decidable statements in Real Algebra. For each of these decidable statements, we seek for certificates. We provide variations of this theme for different contexts: Positivstellensatz, Nichtnegativstellensatz and real radical ideal membership. In our context, our positivity certificates (or membership certificates) provide proof when achieved. We also find plenty of novel numerical evidence substantiating our conjectures.

Contents

Chapter 1

Introduction

Table 1.1 provides a subset[1] of known hypotheses under which Conjecture 0.1 is true[2]. Further history can be found in [34] [17] and the references contained therein. Prior works conceptualize our conjecture in terms of extremal polynomials, namely [58][49] and extensions found in [34], wherein we find language of optimization and real algebra. Prior work, including excellent works of the late Julius Borcea[3], also extend our conjecture to Borcea's $p - variance$ conjectures (BPVC) [34, Conjecture 1].

Our primary novel contribution is that our scripts explicitly translate our conjectures whereby we apply techniques of real algebra [61][47][9][5][21][40][39]. Especially our generating scripts, given a choice of fixed degree, translate our various conjectures as inputs into algorithms of real algebra previously yet applied in the context of our conjectures. Thus, we consider our meager contributions as extensions of [58] [49] [34] and know that we stand upon the shoulders of giants . Historically, especially circa 2011-2012, we focused first on optimization. A priori, these optimization scripts provide both an opportunity for counterexample search *or* numerical certification of global optimality. Our tools provided numerical certificates which, while providing substantial evidence, does

[1]We must stress that we present here only a selection, a strict subset, of prior work. For any omissions, please excuse – the complement of this selection also contains merit.

[2]Our condolences to the family, friends and colleagues of Blagovest Sendov (February 8^{th}, 1932-January 19^{th}, 2020).

[3]Our condolences to friends, family and colleagues of Julius Borcea.

Condition	Author[s]	Year	Ref.
$n = 3$	Brannan	1968	[15]
$n = 4$	Phelps and Rodriguez	1972	[58]
$n = 5$	Meir and Sharma	1969	[48]
$n = 6$	Borcea or Katsoprinakis	1994	[11][33]
$n = 7$	Borcea or Brown	1996	[12][16]
$n \leq 8$	Brown and Xiang	1999	[17]
If root on boundary of unit disk, then it contains c.p. within unit distance.	Rubinstein	1968	[65]
Within a distance of 1.08006, there is a c.p.	Bojanov, Rahman, and Szynal	1985	[10]
As $n \to \infty$, the above bound tends to 1.			ibid
Within a distance of 1.075, there is a c.p.	Rahman and Schmeisser	2002	[62]
If a polynomial has precisely three distinct roots	Saff and Twomey	1971	[66]

Table 1.1: A Selection of Prior Work

not provide proof[4]. These explorations and their conclusions led to an ideal description and searches for analogous certificates within the context of real radical ideals, namely: certificates of real radical ideal membership. In the context of real radical ideal membership, we successfully found closed forms. Note that the search for certificates is not new and dates at least as far back as Hilbert's nullstellensatz and in modern times can be traced to [36] [5]

In Chapter 3, we translate Conjecture 0.1 into a language of Real Algebra, c.f. Conjecture 3.1. For $n = 2$, we obtained a certificate of real radical membership, 3.4. For $n > 2$ we obtained systems of equations which specify certificates [69].[6]

In Chapter 4, we translate Conjecture 2.3 into a language of semialgebraic sets, real-valued optimization, and semidefinite programming (SDP) whereby we utilize work built on foundations of Moment Problems, Linear Algebra and Operator Theory. We obtain numerical Positivstellensatz certificates[7]. Remarkably, for $9 < n < 14$, whenever we terminate the computation, we obtain roots of unity (or rotations thereof). Note that

[4]Proof requires a closed form and our numerical certificates are in machine double precision.

[5]We translated a manuscript of [36].

[6]As we noted in our acknowledgements: with some disappointment we report that forms known to exist where prior work has proved our conjecture ($3 \leq n \leq 8$) still evade our explicit characterization of them. Though, in a spirit of exploration, our silver lining is that here the adventure continues. Scripts [69] are provided which lay the groundwork for finding these certificates of real radical membership.

[7]Within tolerance of machine error.

this agrees with known optima in the cases where Sendov's conjecture has been proved true and conjectured optima in the cases where Conjecture 0.1 has not yet been resolved. I.e., SDP provides numerical evidence for Sendov's conjecture being true[8][9][10][11][12] .

in Chapter 5, we provide applications of Rouche's theorem. In Appendix A.4, we include a draft, from 2012,[13][14][15][16]which includes figures containing illustrative examples. Regarding 2012, we include errata in Appendix A.6, our defense slides from May 2012 [67] in our Supplementary Materials[17]. Also in our Supplementary Materials are two software repositories containing scripts [69] and generating scripts [68] of our translations.

---

[8]Our SDP work touched $2 \leq n \leq 14$.

[9]For $n > 14$, an opportunity for counterexample search.

[10]Here again, our scripts [68] provide groundwork for further exploration.

[11]Our scripts were analogously adjusted to compute in the context of BPVC. Thus, we can provide, for some p numerical evidence for Borcea's $p - variance$ conjectures.

[12]Our experiments mostly focused on $p = 2$.

[13]There are a number of gems here in the 2012 version which deserve a brief mention but overall, there was too much to reproduce it all here in the 2020 version.

[14]it is remarkable that our coding provided such high precision for numerical root computations. Look at page 42 of the 2012 draft. This alone warrants a separate line of inquiry.

[15]The ACOI "path toward proof" or "line of inquiry" (c.f. pages viii,23-26 of 2012 draft) was discussed as a "pullback" during the May 2012 defense and later, I was delighted to read [59] which answers some of questions related to ACOI. My conclusion is that such a line of inquiry proves difficult given the lack of simple connectivity of the pullback but I also conclude that there are some higher powered mathematicians out there who can better bring the ACOI approach with respect to our conjectures to its rightful ends.

[16]Similarly, we drop the reference to our "following critical point that comes along with the root" (Conjecture 2.3.2 on page 13 of 2012 draft) as a "path to proof" which is related to Chapter 5 of this 2020 version. In short, we drop it because too many bifurcations occur. Similarly, we do not provide the details of our applications of Cylindrical Algebraic Decomposition algorithms to our conjectures (which also relates to the Voronai partition idea of the 2012 draft) because the details are messy for the fact the space of root configurations and their corresponding critical point configurations is unwieldy.

[17]which include 'animations' of the dynamics discussed in Chapter 5 amongst other additional material

Chapter 2

Notation and Preliminary Results

2.1 Preliminaries

2.1.0.1 Complex Univariate Polynomials

Let $\mathbb{C}[z]$ be the ring of univariate complex polynomials. Let $f(z) \in \mathbb{C}[z]$ be of degree n. By the Fundamental Theorem of Algebra, $f(z)$ has n roots, $Z = \{z_k\}_{k=1}^{n}$, where z_k are not necessarily distinct. Then we can express $f(z)$ in at least two ways

$$f(z) = \alpha \prod_{\ell=1}^{n} (z - z_\ell) = \alpha \sum_{j=0}^{n} \alpha_j z^j \tag{2.1}$$

where $\alpha_n = 1$.

2.1.0.2 Derivative and Critical Points

The derivative of f is

$$\frac{\partial f}{\partial z} = f'(z) = \alpha \sum_{k=1}^{n} \left(\prod_{\ell=1, \ell \neq k}^{n} (z - z_\ell) \right) = \alpha \sum_{j=1}^{n-1} j \alpha_j z^{j-1} \tag{2.2}$$

Again, applying the Fundamental Theorem of Algebra, $f'(z)$ has $(n-1)$ roots, $W = \{w_\ell\}_{\ell=1}^{n-1}$

$$f'(z) = \beta \prod_{\ell=1}^{n-1} (z - w_\ell) = \beta \sum_{j=0}^{n-1} \beta_j z^j \tag{2.3}$$

We call W the critical points of f.

2.1.1 Unsymmetrized Hausdorff Distance

Let $\mathbb{D}$ be the closed unit disk,

$$\mathbb{D} = \{z \in \mathbb{C} : |z| \leq 1\}.$$

Let $V(f)$ denote the variety of f,

$$V(f) = \{z : f(z) = 0\}.$$

Define the *unsymmetrized Hausdorff distance* ($u\mathcal{H}d$) of f,

$$\Delta(f) \equiv \Delta(Z) \equiv \mathcal{H}(V(f), V(f')) \equiv \max_{z \in Z} \left(\min_{w \in W} |z - w| \right). \tag{2.4}$$

Proposition 2.1 (Continuity of $u\mathcal{H}d$)

As a function,

$$\Delta : \mathbb{C}^n \to \mathbb{R}^+ \cup \{0\}$$

$u\mathcal{H}d$ is continuous.

We can restate Conjecture 0.1 as follows:

Conjecture 2.2 (Sendov's 1958 Conjecture stated in terms of $u\mathcal{H}d$)

If F is a univariate complex polynomial of degree $n \geq 2$ with all roots contained in the closed unit disk, $V(F) \subset \mathbb{D}$, then $\Delta(F) \leq 1$.

2.1.2 Sendov's Conjecture for Fixed Degree (SCFD)

Notation. Fix n where n is an integer greater than or equal to 2. Let $\mathbb{C}_n[z]$ denote the set of all polynomials of degree n or less.

We decompose Conjecture 2.2 into a countable collection of conjectures:

Conjecture 2.3 ($SCFD_n$)
If $F \in \mathbb{C}_n[z]$ and $V(F) \subset \mathbb{D}$, then $\Delta(F) \leq 1$.

We have a number of semialgebraic descriptions, optimization problems, and ideals which describe $SCFD_n$. Indeed, in these terms, $SCFD_n$ is a real decision problem and, thus, decidable.

2.1.3 Roots of Unity

For the roots of $G(z) = z^n - 1$,

$$\Delta(G) = \Delta\left(\{e^{k\pi i/n}\}_{k=1}^{n}\right) = 1.$$

Thus far, roots of unity and rotations thereof are the only known optimal[1] [2] polynomials with respect to $u\mathcal{H}d$ which satisfy the hypothesis of Conjecture 0.1, thus we can restate, as found in [34]:

Conjecture 2.4 (Sendov's 1958 Conjecture stated in terms of roots of unity)
For polynomials satisfying the hypotheses of Conjecture 0.1, optimal polynomials with respect to the $u\mathcal{H}d$ are roots of unity, or rotations thereof.

[1] We derive statements in optimization following any particular conjecture of ours. Sometimes our objective will be to minimize (i.e. minimization programs, sometimes abbreviated min) in which we seek for infimum, denoted inf; other times our objective will be to maximize (i.e. maximization programs, sometimes abbreviated max) in which we seek for supremum, denoted sup.

[2] Because our domains of our optimization programs will be polynomials or linear spaces representing polynomials, an element of *argmax* or *argmin* will be referred to as an *optimial polynomial with respect to* a given norm, metric, real-valued function, or function whose codomain has an ordering...or an *optimal polynomial* when our functions are implied. Especially when our optimization program is related to convex analysis, we will also refer to our set of *argmax* or *argmin* as *extremal polynomials*.

$$Z = \{e^{\theta_0 + k\frac{2\pi}{n}i} : 0 \leq k \leq n-1\}$$

2.2 Elementary Symmetric Polynomials

2.2.0.1 Products of monic affine terms as sums of elementary symmetric polynomials

For convenience of notation, let $T = \{t_1, t_2, \ldots, t_m\} = \{t_j\}_{j=1}^m$. Label the following sets:

$$\begin{aligned}
T_0 &= \emptyset \\
T_1 &= \{t_1\} \\
T_2 &= \{t_1, t_2\} \\
&\vdots \\
T_k &= \{t_1, t_2, \ldots, t_k\} \\
&\vdots \\
T_{m-1} &= T \backslash \{t_m\} \\
T_m &= T
\end{aligned}$$

Definition 2.1. Elementary symmetric polynomials Let $\sigma_\ell(T)$ be the sum of all distinct products of ℓ elements of T, also know as the elementary symmetric polynomial of order ℓ of T. By definition,

$$\begin{aligned}
\sigma_0(T) &= 1 \\
\sigma_1(T) &= \sum_{k=1}^{m} t_k = t_1 + t_2 + \ldots + t_m \\
\sigma_m(T) &= \prod_{k=1}^{m} t_k = t_1 t_2 \cdots t_m
\end{aligned}$$

We can recursively define σ_ℓ as follows:

$$\sigma_\ell(T_k) = \begin{cases} 1 & \ell = 0 \\ \sigma_\ell(T_{k-1}) + t_k\sigma_{\ell-1}(T_{k-1}) & 0 < \ell < k \\ \prod_{j=1}^{k} t_j & \ell = k \end{cases} \tag{2.5}$$

We provide proof of a classical result.

Proposition 2.5 (Symmetric Polynomials of Roots are, up to sign, Coefficients of Polynomial)
Let

$$\gamma_j = (-1)^{(n+j)}\sigma_{(m-j)}(T) \tag{2.6}$$

then

$$\prod_{\ell=1}^{m}(z - t_\ell) = \sum_{j=0}^{m}\gamma_j z^j$$

Example 2.6 (Example of Proposition 2.5 when $m = 2$).

$$T = \{t_1, t_2\}$$

$$\prod_{j=1}^{2}(z - t_j) = (z - t_1)(z - t_2) = z^2 - (t_1 + t_2)z + t_1t_2$$

$$\begin{aligned} \sigma_1(T) &= t_1 + t_2 \\ \sigma_2(T) &= t_1t_2 \end{aligned}$$

Example 2.7 (Example of Proposition 2.5 when $m = 3$).

$$T = \{t_1, t_2, t_3\}$$

$$\prod_{j=1}^{3}(z-t_j) = (z-t_1)(z-t_2)(z-t_3) = z^3 - (t_1+t_2+t_3)z^2 + (t_1t_2+t_1t_3+t_2t_3)z - t_1t_2t_3$$

$$\begin{aligned}
\sigma_1(T) &= t_1+t_2+t_3 \\
\sigma_2(T) &= t_1t_2+t_1t_3+t_2t_3 \\
\sigma_3(T) &= t_1t_2t_3
\end{aligned}$$

Proof of Proposition 2.5. We prove by induction. Our base cases are the above examples.

Induction step:

$$\prod_{k=1}^{m}(z-t_k)$$

Pull out the factor, $(z-t_m)$,

$$= (z-t_m)\prod_{k=1}^{m-1}(z-t_k)$$

Use induction hypothesis

$$= (z-t_m)\left(\sum_{p=0}^{m-1}(-1)^{(m-1-p)}\sigma_{(m-1-p)}(T_{(m-1)})z^p\right)$$

Distribute

$$= \left(\sum_{p=0}^{m-1}(-1)^{(m-1-p)}\sigma_{(m-1-p)}(T_{(m-1)})z^{(p+1)}\right)$$

$$- t_m\sum_{p=0}^{m-1}(-1)^{(m-1-p)}\sigma_{(m-1-p)}(T_{(m-1)})z^p$$

Pull one term out of each sum

$$= z^m + \left(\sum_{p=0}^{m-2}(-1)^{(m-1-p)}\sigma_{(m-1-p)}(T_{(m-1)})z^{(p+1)}\right)$$

$$+ \left(-t_m\sum_{p=1}^{m-1}(-1)^{(m-1-p)}\sigma_{(m-1-p)}(T_{(m-1)})z^p\right)$$

$$+ (-1)^m t_m\sigma_{(m-1)}(T_{(m-1)})$$

Relabel the first sum with substitution $p_{new} = p_{old} + 1$

...and recognize last term is, up to sign, $\prod_{k=1}^{m} t_k$

$$= z^m + \left(\sum_{p=1}^{m-1} (-1)^{(m-p)} \sigma_{(m-p)}(T_{(m-1)}) z^p\right)$$
$$+ \left(-t_m \sum_{p=1}^{m-1} (-1)^{(m-1-p)} \sigma_{(m-1-p)}(T_{(m-1)}) z^p\right)$$
$$+ (-1)^m \sigma_m(T)$$

Collect like terms between the sums

$$= z^m + \sum_{p=1}^{m-1} (-1)^{(m-p)} \left[\sigma_{(m-p)}(T_{(m-1)}) + t_m \sigma_{m-1-p}(T_{m-1})\right] z^p$$
$$+ (-1)^{(m-p)} t_m \sigma_{(m-1)}(T_{(m-1)})$$

Recognize recursive definition of σ, equation 2.5

$$= z^m$$
$$+ \sum_{p=1}^{m-1} (-1)^{(m-p)} \sigma_{(m-p)}(T) z^p$$
$$+ (-1)^m \sigma_m(T)$$

Collect terms in sum.

$$= \sum_{p=0}^{m} (-1)^{(m-p)} \sigma_{(m-p)}(T) z^p$$

□

2.2.1 Sufficient equations relating critical points and roots

Proposition 2.8 (Implicit Relationship Between Roots and Critical Points)

$$\ell \sigma_{(n-\ell)}(Z) = n \sigma_{(n-\ell)}(W) \tag{2.7}$$

for $1 \leq \ell \leq n$.

Remark 2.1 (Defining Critical Points from Roots). For $\ell < n$, the set of nontrivial equations 2.7 implicitly define the critical points in terms of the roots.

Proof of Proposition 2.8. By equating coefficients from equations 2.2 and 2.3 we obtain the following equations[3]

$$\begin{aligned}
\alpha \cdot \alpha_1 &= \beta \cdot \beta_0 \\
\alpha \cdot 2\alpha_2 &= \beta \cdot \beta_1 \\
&\vdots \\
\alpha \cdot k\alpha_k &= \beta \cdot \beta_{k-1}
\end{aligned}$$

With the leading terms providing the equation

$$\alpha n = \beta$$

with which we can substitute to obtain:

$$\ell \alpha_\ell = n\beta_{\ell-1} \tag{2.8}$$

for $1 \leq \ell \leq n$. Now consider equation 2.6 for α_ℓ

$$\alpha_\ell = (-1)^{(n+\ell)} \sigma_{(n-\ell)}(Z) \tag{2.9}$$

Similarly, while recalling that the cardinality of W is $(n-1)$,

$$\beta_\ell = (-1)^{(n-1)+\ell} \sigma_{(n-1)-\ell}(W)$$

Thus,

$$\beta_{\ell-1} = (-1)^{(n-1)+(\ell-1)} \sigma_{(n-1)-(\ell-1)}(W)$$

$$\beta_{\ell-1} = (-1)^{(n+\ell)} \sigma_{(n-\ell)}(W) \tag{2.10}$$

By substituting 2.9 and 2.10 into 2.8 we obtain 2.7 for $1 \leq \ell \leq n$. $\square$

Corollary 2.9

[3] We could prove a similar result starting with monic polynomials, i.e. requiring $\alpha = 1$ in 2.1, however we keep this more general (but admittedly more cumbersome notation) for geometric reasons (specifically to allow "One root at 1" to remain expressible within the context of 2.1).

By Proposition 2.8, specifically by plugging Equation 2.9 into Equation 2.2 and then reversing index labelling, we can express $f'(z)$ as

$$\frac{\partial f}{\partial z} = \alpha \sum_{\ell=0}^{n-1} (n-\ell)\sigma_\ell(Z)(-1)^{n-\ell} z^{n-\ell-1} \tag{2.11}$$

2.2.2 Derivative of Elementary Symmetric Polynomial

We include the following equation, relatively easy to prove. We employ it in our explorations of Cylindrical Algebraic Decompositions, though we won't detail those experiments and will save comments regarding them for our concluding chapter. We include it here because it might still be a valuable piece of our puzzle, for posterity.

$$\frac{\partial \sigma_\ell(\{z_j\}_{j=1}^{n})}{\partial z_n} = \sigma_{\ell-1}(\{z_j\}_{j=1}^{n-1}) \tag{2.12}$$

2.3 Geometry

2.3.1 Scaling and Rotations

Let $\kappa \in \mathbb{C}$, with polar decomposition, $\kappa = re^{i\theta}$ so that multiplying by κ corresponds to a planar translation consisting of scaling by r and a rotation by θ.

Proposition 2.10 (Multiplying the root set by κ corresponds to multiplying the critical point set by κ)

If we translate,

$$\tilde{Z} = \kappa Z = \{\kappa z_1, \kappa z_2, \ldots, \kappa z_n\}$$

then the critical point set $\tilde{W}$ corresponding to the root set $\tilde{Z}$ is

$$\tilde{W} = \kappa W = \{\kappa w_1, \kappa w_2, \ldots, \kappa w_{n-1}\}$$

Proof. Proof of Proposition 2.10 For each $1 \leq \ell \leq n-1$, Equation 2.7 becomes

$$(n-\ell)\sigma_\ell(\tilde{Z}) = \kappa^\ell (n-\ell)\sigma_\ell(Z) = \kappa^\ell n \sigma_\ell(W) = n\sigma_\ell(\tilde{W})$$

To elaborate, we develop a commutative diagram.

Let $\mathbb{C}_s^n = \mathbb{C}^n/\{\text{up to ordering}\}$. Let $\varphi : \mathbb{C}_s^n \to \mathbb{C}_s^{n-1}$ be the canonical map defined by application of Fundamental Theorem of Algebra and canonical differential operator, so that $\varphi(Z) = W$. Respectively define maps $\psi_n : \mathbb{C}_s^n \to \mathbb{C}_s^n$ and $\psi_{n-1} : \mathbb{C}_s^{n-1} \to \mathbb{C}_s^{n-1}$ so that each corresponds to multiplying a point-set by $\kappa \in \mathbb{C}$, with

$$\psi_n(Z) = \tilde{Z}$$
$$\psi_m(W) = \tilde{W}$$

The following are equivalent:

- $\begin{tikzcd} Z \arrow[r, "\varphi"] \arrow[d, "\psi_n"'] & W \arrow[d, "\psi_{n-1}"] \\ \tilde{Z} \arrow[r, "\varphi"'] & \tilde{W} \end{tikzcd}$ is a commutative diagram

- $\{\sigma_\ell(\tilde{W}) = \sigma_\ell(\tilde{W}')\}_{1\leq \ell \leq n-1} \Rightarrow \tilde{W} = \tilde{W}'$

We address in our diagram our ability to recover a pointset $\tilde{W}$, up to ordering, from the values of its corresponding ordered set of values of elementary symmetric polynomials evaluated on $\{\sigma_\ell(\tilde{W})\}_{0\leq \ell \leq n-1}$. [4][5] □

Corollary 2.11

Rotating zero set by angle θ rotates the critical point set by angle θ.

Following Proposition 2.10, we have

[4]That we can recover pointset $A_n = \{a_1, \ldots, a_n\}$ from coefficients of $p(z) = \prod_{k=1}^n (z-a_k)$ follows from induction, i.e repeated application of the Fundamental Theorem of Algebra ala $p(z)/(z-a_n) = \prod_{k=1}^{n-1}(z-a_k)$.

[5]Thank you to our proofreader, Travis Waddington, for discussions regarding this proof.

Corollary 2.12 (Scaling and $u\mathcal{H}d$)

$$\Delta(\tilde{Z}) = \mathcal{H}(\tilde{Z}, \tilde{W}) = \max_{z \in Z} \min_{w \in Z} |\kappa z - \kappa w| = \max_{z \in Z} \min_{w \in Z} |\kappa||z - w| = |\kappa|\Delta(Z) \tag{2.13}$$

Remark 2.2. Specifically, $u\mathcal{H}d$ is rotationally invariant. [6]

2.3.2 Centroids

Equating the second-most leading coefficient of each expression in Equation 2.2 is equivalent to letting $\ell = n - 1$ in Proposition 2.8 and we restate it here in an alternate form:

Corollary 2.13 (Centroid of Roots is Centroid of Critical Points)

$$\frac{1}{n}\sum_{k=1}^{n} z_k = \frac{1}{n-1}\sum_{j=1}^{n-1} w_j$$

In terms of point masses, the centroid can be considered the first moment. In other words, our corollary states: the first moment of a set of roots is equal to the first moment of their critical points. Thus, Proposition 2.8 provides higher order moment relationships between a set of roots and its corresponding set of critical points.

2.3.3 Each Critical Point is a Convex Combination of the Roots

Lemma 2.14 (Result upon Considering Logarithmic Derivative)

If w is a critical point which is not a root,

$$0 = \sum_{k=1}^{n} \frac{1}{w - z_k} \tag{2.14}$$

[6] As an imaginative aside: if we prove $u\mathcal{H}d$ is monotonic with respect to some symmetry metric for which the roots of unity are optimal (perhaps some angular symmetry metric), then we achieve Conjecture 2.2. This idea, in some form or another, was discussed briefly and independently, on two separate occasions: once with Mihai and once with Pablo Parrilo. Especially in protest to the mess of indexing to which we subject the reader in our scripts. Our apologies, in advance to you, dear reader, for the rigmarole ...but the indexing the document does reflect our scripts and our scripts are unit tested and our readers can reproduce our findings with the materials provided by us, so, well, it's defensible.

Proof.

$$\frac{f'(z)}{f(z)} = \sum_{k=1} \frac{1}{z - z_k}$$

Let w be a critical point of f which is not a root (i.e. no multiple roots of f). Then $f'(w) = 0$ and our results follows from the equation above, our computation of a logarithmic derivative of f. $\square$

Remark 2.3. We did not explicitly write

$$\frac{\partial log(f(z))}{\partial z} = \frac{f'(z)}{f(z)}$$

above because it would be necessarily to choose a region on analyticity (taking care of poles at roots) and an analytic branch. C.f. "branch" in [1] [4],[25] [20]

Proposition 2.15 (Every critical point is a convex combination of zeros.)
Let w be a critical point. Then

$$w = \sum_{k=1}^{n} \lambda_k z_k \tag{2.15}$$

with $\lambda_k \geq 0$ and $1 = \sum \lambda_k$.

Proof. If w is a multiple root of f, then our result follows trivially. Otherwise,

Start by taking the complex conjugate of both sides of Equation 2.14

$$0 = \sum_{k=1}^{n} \frac{1}{\overline{w - z_k}}$$

Multiply each term by 1.

$$= \sum_{k=1}^{n} \frac{1}{\overline{w - z_k}} \frac{w - z_k}{w - z_k} = \sum_{k=1}^{n} \frac{w - z_k}{|w - z_k|^2}$$

Distribute and then separate sums.

$$= \sum_{k=1}^{n} \frac{w}{|w - z_k|^2} - \frac{z_k}{|w - z_k|^2} = \left(\sum_{k=1}^{n} \frac{w}{|w - z_k|^2} \right) - \left(\sum_{k=1}^{n} \frac{z_k}{|w - z_k|^2} \right)$$

Bring each sum to a side, and then factor out w

$$w\sum_{k=1}^{n}\frac{1}{|w-z_k|^2}=\sum_{k=1}^{n}\frac{z_k}{|w-z_k|^2}$$

Used a different dummy variable in the left sum, divide by it...

$$w=\frac{1}{\sum_{\ell=1}^{n}\frac{1}{|w-z_\ell|^2}}\sum_{k=1}^{n}\frac{z_k}{|w-z_k|^2}$$

...and then and then distribute it across the sum.

$$w=\sum_{k=1}^{n}\frac{1/|w-z_k|^2}{\sum_{\ell=1}^{n}\frac{1}{|w-z_\ell|^2}}z_k$$

Our result follows by letting

$$\lambda_k=\frac{1/|w-z_k|^2}{\sum_{\ell=1}^{n}1/|w-z_\ell|^2} \tag{2.16}$$

□

2.3.4 Roots on Boundary: A Result of Rubinstein

In [65] we find

Theorem 2.16 (Rubinstein, 1968)

If a root is on boundary of $\mathbb{D}$, then a critical point is within unit distance.

Proof. Let $f\in\mathbb{C}[z]$ satisfy the hypothesis of Conjecture 0.1. Furthermore, assume f has a root, labelled z_1, such that $|z_1|=1$. If z_1 is a root of multiplicity greater than 1, then we are done. Otherwise, without loss of generality, assume $z_1=1$ and $z_k\neq 1$ for $k=2,3,\ldots,n$ and $f'(1)=1$.

Toward contradiction, assume that no critical point is within unit distance of z_1. Thus, $f'(z+1)$ will have no root for $|z|<1$. Consider an analytic expansion

$$\begin{aligned}(f'(z+1))^{1/(n-1)}&=1+\text{higher order terms}\\&=1+z(\ldots)\\&=1-z\cdot h(z)\end{aligned}$$

Thus, for $|z| < 1$, $f'(z+1) = (1 - zh(z))^{n-1}$ where $h(z)$ is analytic in the open unit disc and $|h| < 1$.

By differentiation, $f''(1) = (1-n)h(0)$.

The polynomial $Q(z) = f(z)/(z-1)$ satisfied $Q(1) = f'(1) = 1$ and $2Q'(1) = f''(1)$, so that

$$|Q'(1)| < \frac{n-1}{2}$$

However, consider a logarithmic derivative of Q,

$$Q'(1) = \sum_{k=2}^{n} \frac{1}{1 - z_k}$$

and

$$|z_k| \leq 1 \Rightarrow Re[1/(1 - z_k)] \geq \frac{1}{2}$$

thus,

$$Re[Q'(1)] \geq \frac{n-1}{2}$$

□

Remark 2.4. Another presentation of proof of 2.16 is provided c.f. pg. 5 Appendix A.4.

2.3.5 "One root at 1"(OaO)

Within the context of Conjecture 2.2, Equation 2.13 implies that any optimal configuration of roots with respect to $u\mathcal{H}d$ will have at least one root on the boundary. Furthermore, by Remark 2.2 following Corrollary 2.12 we can rotate our root set so that at least one root is equal to 1. Thus, Conjecture 2.2 is equivalent to the following conjecture.

Conjecture 2.17 ("One root at 1"Version of Sendov's Conjecture)
If F is a univariate complex polynomial of degree $n \geq 2$ with all roots contained in the

closed unit disk, $V(F) \subset \mathbb{D}$, and such that at least one root is 1, then $\Delta(F) \leq 1$.

Chapter 3

Ideal Descriptions

3.1 Introduce Real Variables

In this section we introduce real variables

$$\{a_k, b_k, s_k\}_{k=1}^{n} \text{ and } \{c_\ell, d_\ell, t_\ell\}_{\ell=1}^{n-1}$$

which will be considered members of a real polynomial ring. We will take the hypotheses of Conjecture 2.2 and translate them into polynomial generators of an ideal, thereby obtaining an ideal description.

3.1.0.1 Cartesian Representation of Roots

For $1 \leq k \leq n$, let the real and imaginary parts of z_k be respectively, a_k and b_k,

$$z_k = a_k + b_k i$$

A hypothesis of Conjecture 0.1 is that the roots are contained within the closed unit disk. The inequality constraint

$$|z_k| \leq 1$$

can be described as an equation by introducing a real slack variable, s_k.

$$|z_k|^2 + s_k^2 = 1$$

from which we obtain

$$a_k^2 + b_k^2 + s_k^2 - 1 \tag{3.1}$$

as a polynomial generator. Here, we obtain n generators.

3.1.0.2 Cartesian Representation of Critical Points

For $1 \leq \ell \leq n-1$, let the real and imaginary parts of w_ℓ be respectively, c_ℓ and d_ℓ,

$$w_\ell = c_\ell + d_\ell i$$

3.1.1 Implicitly define relationship between critical points and roots

We recursively construct the elementary symmetric polynomials and equate them by equation 2.7 to obtain:

$$(n-\ell)\sigma_\ell(Z) - n\sigma_\ell(W) \tag{3.2}$$

and then take the real and imaginary parts of the above expression, for $1 \leq \ell \leq n-1$. Here, we obtain $2(n-1)$ additional generators.

3.1.2 Assert $u\mathcal{H}d \geq 1$.

Without loss of generality, we may pick one root, say $z_1 = a_1 + b_1 i$ and then assert that all critical points are at least a unit distance away, then for each critical point, w_ℓ, we have an equation

$$(a_1 - c_\ell)^2 + (b_1 - d_\ell)^2 = 1 + t_\ell^2$$

where we introduce slack variable, t_ℓ. Here, we obtain $n-1$ additional generators.

$$(a_1 - c_\ell)^2 + (b_1 - d_\ell)^2 - 1 - t_\ell^2 \tag{3.3}$$

3.2 An Ideal Description of Sendov's Conjecture

Definition 3.1 (Ideal Description of Sendov's Conjecture)**.** Define I_n to be the ideal generated by the sets of unit disk constraints (3.1), the real and imaginary parts of the symmetric polynomial constraints (3.2), and the Hausdorff distance constraints (3.3).

$$
\begin{aligned}
I_n = \langle & \\
& \{ \\
& a_1^2 + b_1^2 + s_1^2 - 1, \\
& a_2^2 + b_2^2 + s_2^2 - 1, \\
\vdots & \\
& a_n^2 + b_n^2 + s_n^2 - 1, \\
& \\
& Re[(n-1)\sigma_1(Z) - n\sigma_1(W)], \\
& Im[(n-1)\sigma_1(Z) - n\sigma_1(W)], \\
& Re[(n-2)\sigma_2(Z) - n\sigma_2(W)], \\
& Im[(n-2)\sigma_2(Z) - n\sigma_2(W)], \\
\vdots & \\
& Re[\sigma_{n-1}(Z) - n\sigma_{n-1}(W)], \\
& Im[\sigma_{n-1}(Z) - n\sigma_{n-1}(W)], \\
& \\
& (a_1 - c_1)^2 + (b_1 - d_1)^2 - 1 - t_1^2, \\
& ((a_1 - c_2)^2 + (b_1 - d_2)^2 - 1 - t_2^2, \\
\vdots & \\
& (a_1 - c_{n-1})^2 + (b_1 - d_{n-1})^2 - 1 - t_{n-1}^2 \\
& \} \\
\rangle &
\end{aligned}
$$

Notation. Respectively label the polynomials described above as follows:

$$\begin{aligned}
U_k &= a_k^2 + b_k^2 + s_k^2 - 1,\\
S_{R_\ell} &= Re[(n-\ell)\sigma_\ell(Z) - n\sigma_\ell(W)],\\
S_{I_\ell} &= Im[(n-\ell)\sigma_\ell(Z) - n\sigma_\ell(W)],\\
D_\ell &= (a_1 - c_\ell)^2 + (b_1 - d_\ell)^2 - 1 - t_\ell^2
\end{aligned}$$

where the letters U,S, and D respectively correspond to "unit circle," "symmetric polynomials," and "distance".

Notation. Let $G_{n,0}$ be the set of generators for I_n. That is,

$$G_{n,0} = \{U_k\}_{1\le k\le n} \cup \{S_{R_\ell}, S_{I_\ell}, D_\ell\}_{1\le \ell\le n-1}$$

and

$$I_n = \langle G_{n,0}\rangle$$

3.2.1 A conjecture in terms of I_n which implies $SCFD_n$

Notation. Let K be the polynomial ring

$$K = \mathbb{R}[a_1,\dots,a_n,b_1,\dots,b_n,c_1,\dots,c_{n-1},d_1,\dots,d_{n-1},s_1,\dots,s_n,t_1,\dots,t_{n-1}]$$

and let Σ denote the finite sums of squares (SOS) polynomials which are expressible in the polynomial ring. That is,

$$\Sigma = \left\{ \sum f_i^2 \,\middle|\, f_i \in K \right\}$$

As found in in the field of Real Algebraic Geometry, let $\sqrt[\mathbb{R}]{}$ denote the real radical of an ideal.

In our context, for ideal $J \subset K$, let

$$\sqrt[\mathbb{R}]{J} = \left\{ f | (((\exists) m \in \mathbb{N}) \bigwedge ((\exists) \sigma \in \Sigma) \bigwedge ((\exists) g \in J)) \text{ such that } f^{2m} = -\sigma + g \right\}$$

Further, if J is expressed as the ideal generated by a finite set of generators g_ℓ (in our case, the number of generators of I_n is $4n-3$), then there will be polynomials $k_j \in K$ such that

$$g = \sum k_\ell g_\ell$$

When the context is clear, then we will call a set

$$\{m, \sigma, \{k_\ell\}\} \in \mathbb{N} \times \Sigma \times (\times K)^{4n-3}$$

a *certificate of real radical ideal membership* of f whenever

$$f^{2m} = -\sigma + \sum k_\ell g_\ell$$

Conjecture 3.1 (Real Radical Membership for fixed degree)

$$t_1 \in \sqrt[\mathbb{R}]{I_n}$$

implies Conjecture 2.3.

Inspired by Conjecture 2.4, a stronger version of Conjecture 3.1 is:

Conjecture 3.2 (Real Radical Membership for fixed degree)

$$\left(\{s_k\}_{k=1}^{n} \cup \{t_j, c_j, d_j\}_{j=1}^{n-1}\right) \subset \sqrt[\mathbb{R}]{I_n}$$

3.2.2 A certificate of real radical ideal membership

Remark 3.1. We know Conjecture 2.3 is true for $2 \leq n \leq 8$. [15] [58] [48] [11] [33] [12] [16] [17]

Therefore, we know for each $2 \leq n \leq 8$, there is an expression of t_1 as an element of the real radical of I_n.

3.2.2.1 $t_1 \in I_2$

For this subsubsection, let $n = 2$.

Let

$$G = c_1^2 + d_1^2 + \frac{s_1^2 + s_2^2}{2} + t_1^2$$

and notice that

$$G = U_1 + U_2 - 2D_1 + (a_1 - a_2 - 2c_1)S_{R_1} + (b_1 - b_2 - 2d_1)S_{I_1} \tag{3.4}$$

Then for the list of generators

$$[U_1, U_2, S_{R_1}, S_{I_1}, D_1]$$

and a list of coefficients (considered elements of K in the module which generates I_2)

$$L = [1, 1, S_{R_1} - 2a_2, S_{I_1} - 2b_2, -2]$$

and the sum of squares

$$\sigma = c_1^2 + d_1^2 + \frac{s_1^2 + s_2^2}{2}$$

we have obtained a certificate of real radical ideal membership of $t_1 \in I_2$, namely

$$\left\{ m = 1, \sigma = c_1^2 + d_1^2 + \frac{s_1^2 + s_2^2}{2}, L = [1, 1, S_{R_1} - 2a_2, S_{I_1} - 2b_2, -2] \right\} \tag{3.5}$$

3.3 Algorithms for searching for certificates of real radical ideal membership

We will start with Algorithm 3.1, which is a relatively high-level abstract algorithm.

Remark 3.2. Note that there are many choices that can be made toward an implementation of such an algorithm. Importantly: which subset of sums of squares and how they are parameterized [60] will determine whether the system of equations in the last line of Algorithm 3.1 is linear or quadratic.

Algorithm 3.1 Search for certificate of Real Radical Ideal Membership

1: Parameterize a subset of finite sums of squares. Label parameterized polynomial g_Σ.

2: Parameterize a subset of an ideal. Label parameterized polynomial g_I.
3: Collect terms of $f^{2m} + g_\Sigma$ and then equate with g_I.
4: Decompose the above equation into a system of equations by collecting coefficients (in terms of parameters only) of each monomial
5: Solve system of equations.

We also find algorithms for real radical ideal membership in [38] [41] [79].

3.3.1 Implementations of Algorithm 3.1

Here we try to say too much at once, which is an indication that this is our current best sense of directions in which to move forward[1] in regards to finding proof certificates for $SCFD_n$. In our supplementary material [69] we provide implementations of Algorithm 3.1.

We present[2] a naive[3][4] implementation in Appendix A.1.

[1] i.e. further work, i.e. directions for further research. We have a "dense tangle of ideas" here that can be separated, given further resources.

[2] Much is there that has been said in private correspondence to our friends and much more can be said regarding going between Algorithm 3.1 and any particular version of an implementation...but we want to honor a commitment to get this dissertation done by May 2020 so we'll skip to one such version.

[3] rather than a superior Linear Matrix Inequality (LMI) approach [60] (there are mathematical depths here, e.g. c.f. research in linear pencils)

[4] We also choose to present this naive approach to emphasize a distinction between the parameters

Remark 3.3. For some implementatins, we purposefully find linear systems to be solved in State 5 by choosing in State 1:

- other parameterizations of sums of squares (for example, some LMI)
- or other subsets of sums of squares (for example sums of squares of only variables, rather than full sums of squares).

Remark 3.4. We choose to present[5] this version of the implementation because it contains a validation example whereby we confirm Equation 3.4 verified.

3.4 Simple, Concise Code for Equation 3.4

We provide[6] the following Sage code which returns True.

```
R.<a1,a2,b1,b2,c1,d1,s1,s2,t1> = PolynomialRing(QQ,9,order='lex')
I=Ideal([
1-a1^2-b1^2-s1^2,\
1-a2^2-b2^2-s2^2,\
\
1*(a1+a2)-2*(c1),\
1*(b1+b2)-2*(d1),\
\
((a2-c1)^2+(b2-d1)^2)-1-t1^2,\
])
g=I.groebner_basis()
g[8]==c1^2+d1^2+1/2*s1^2+1/2*s2^2+t1^2
```

used to parameterize sums-of-squares versus the parameters used to parameterize ideal member. We hope it allows our readers to absorb with a little more ease, the algorithm.

[5] I would rather break this apart by paragraph and provide alternative implementations per paragraph, which is also possible by using interfaces...similar to that of the attached Java package

[6] Because we feel a bit out on a scientific limb.

Other computations, especially those which respect symmetry [70], were also used to compute Gröbner bases[74]. For $n > 2$, it seems necessary to perform a real radical ideal search. In other words, the sums of squares certificates are not necessarily in the Groeber basis.

3.5 Other Certificates in the context of Conjecture 2.17, when $n = 2$

Within the context of Conjecture 2.17, a Gröbner fan [32] contains a number of different sums of squares which provide certificates of real radical membership for t_1. For example,

$$d_1^2 + t_1^2 + t_1^4 + \frac{1}{2}s_1^2 + \frac{1}{2}s_1^2 t_1^2 + \frac{1}{16}s_1^4$$

$$t_1^4 + t_1^2 + \frac{1}{2}s_1^2 + \frac{1}{2}s_1^2 t_1^2 + \frac{1}{16}s_1^4 + d_1^2$$

$$b_1^2 + 4t_1^2 + 4t_1^4 + 2s_1^2 + 2s_1^2 t_1^2 + \frac{1}{4}s_1^4$$

$$t_1^4 + t_1^2 + \frac{1}{2}s_1^2 + \frac{1}{2}s_1^2 t_1^2 + \frac{1}{16}s_1^4 + \frac{1}{4}b_1^2$$

$$s_1^4 + 16t_1^2 + 16t_1^4 + 8s_1^2 + 8s_1^2 t_1^2 + 4b_1^2$$

$$s_1^4 + 16t_1^2 + 16t_1^4 + 8s_1^2 + 8s_1^2 t_1^2 + 16d_2^2$$

Remark 3.5. Notice, however, that:

- these SOS polynomials are all of higher degree, 4, than the degree of 3.5, 2.
- there's reference to b_1 which indicates the existential risk that for $n > 2$ the lack of full symmetry here will make messier coefficients for the real and imaginary parts of roots, as opposed to all their coefficients being 0 in 3.5.

3.6 Exploration of S-Procedure

The S-Procedure is found in the heart of algorithms in Real Algebraic Geometry [5]. It is found in Bucherber's algorithm, e.g. c.f. Chapter 2, Section 7, Theorem 2 of [21].

For the remainder of this subsection, we will fix our monomial ordering to be lexicographical ordering. Thus, the leading terms are:

$$\begin{aligned}
LT(U_k) &= a_k^2 \\
LT(S_{R_\ell}) &= (n-\ell)\prod_{j=1}^{\ell} a_j \\
LT(S_{I_\ell}) &= (n-\ell)b_1\prod_{j=2}^{\ell} a_j \\
LT(D_\ell) &= a_1^2
\end{aligned}$$

We take into consideration Chapter 2, Section 9, Proposition 4 of [21], which in the context of our work, states that if the leading terms are relatively prime, then the $S - polynomial$ will reduce to zero module the set of generators. Thus, if we follow Buchberger's algorithm, then in the initial step we focus on the following sets of pairs:

$$\begin{aligned}
&\{S(U_1, S_{R_j})\} \text{ for } 1 \le j \le n-1 \\
&\{S(U_1, S_{I_j})\} \text{ for } 2 \le j \le n-1 \\
&\{S(S_{R_j}, S_{I_j})\} \text{ for } 2 \le j \le n-1 \\
&\{S(U_1, D_j)\} \text{ for } 1 \le j \le n-1 \\
&\{S(S_{R_j}, D_\ell)\} \text{ for } 1 \le j, \ell \le n-1 \\
&\{S(S_{I_j}, D_\ell)\} \text{ for } 2 \le j \le n-1, 1 \le \ell \le n-1
\end{aligned}$$

We made some progress finding closed forms for the above. We take this approach because to search for real radical ideal membership for each n in hopes to find a generalization of Equation 3.4 for every n. In general, the Gröbner basis techniques, and in particular, Buchberger's algorithm, are doubly exponential in computational complexity.

This is really poor from a computational complexity point of view because computer scientists tend to prefer algorithms that are constant time, logarithmic, linear time, linearithmic (n log n), and rarely will they every put up with anything that's quadratic...let alone exponential. However, for general decision problems in real algebra (those which follow from Tarski-Seidenberg), doubly exponential is the known best. [7]

[7]For example, the Cylindrical Algebraic Decomposition is doubly-exponential for much the same reasons (used, for e.g., to provide global optimization certificates).

Chapter 4

Semidefinite Sets, Positive Polynomials and Conjecture 2.2

In general terms, we wrote a script which accepted as input n in SCFD and generated a decidable program.[1][2][3] As another instance of decision theory in Real Algebra, we can describe Sendov's conjecture in terms of a constrained polynomial optimization problem. Recall in Section 3.2 we considered unit disk constraints, symmetric polynomial constraints, and Hausdorff distance constraints when translating Sendov's conjecture into a language of Real Algebra. We proceed again, this time in the language of real-valued, constrained polynomial optimization.

We then apply techniques of semidefinite programming (insert list of SDP backends we used) especially focusing on the interfaces provided by Gloptipoly, YALMIP, and SOSTOOLs. Gloptipoly is software founded on moment theory [42][72][8][22] and positive

[1]For semidefinite sets, c.f. [54] [76] [52] [51] [60] [64] [47] [40] [61] [71] [18] [27] and the references therein.

[2]For Positive Polynomials, c.f. [47] [40][61][18][24][6][7][29]

[3]We also experimented with Cylindrical Algebraic Decomposition and resultants [9]. Like other decision algorithms for general decision problems in Real Algebra, the complexity analysis is doubly exponential [5][81]. Also, for reasons that can be pointed to in the dynamics section, when trace back "follow the critical point that goes out with the root, z_n" (i.e., let R decrease from sufficiently large back to R_n) we notice many opportunities for bifurcation, which would make CAD quite messy in this context. Similarly, for Voronai techniques.

polynomials [40][47][61]. YALMIP [44][43] is an acronym for "Yet Another Linear Matrix Inequality Program." [4]

SOSTOOLs is a Sums of Squares program [53] which provides for search of numerical positivity certificates and also relates to a duality between sums of squares and moments.

The optimization problem

$$\sup_{F \in \mathbb{C}[z]_n} \Delta(F)$$

can be translated into a real-valued, constrained, real polynomial optimization problem. In particular, we utilize a technique communicated to us by our advisor circa 2011 [3] [39] whereby we express

$$\max(a, b) = \frac{a + b + |a - b|}{2} \tag{4.1}$$

$$\min(a, b) = \frac{a + b - |a - b|}{2} \tag{4.2}$$

and note that we can express $|a - b|$ by lifting (i.e. as a semidefinite set by introducing variable c)

$$\begin{aligned} c^2 &= (a - b)^2 \\ c &\geq 0 \end{aligned}$$

Our optimization programs, in particular, can be construed as a counterexample search or, alternatively, numerical positivity certificates. Our numerical positivity certificates provide numerical evidence for 2.2, though our numerics, based on machine precision, lack the certainty of 3.4.

Here, we present a translation of Conjecture 2.3 into the language of semidefinite sets. Because of resource limitations, we provide an example of only one flavor of our scripts. For further details, please examine our ConstraintGenerator projects in Supplementary

[4]For more on Linear Matrix Inequalities (LMI) [28] and related topics, c.f. [26] [74] [77] [78]

Materials [68].

4.1 One Root Fixed

Our context for this section is Conjecture 2.17. Without loss of generality (WLOG), we can label $z_1 = 1 + 0i$ and by Theorem 2.16, we can conclude that z_1 has a critical point within unit distance. Thus, we maximize $u\mathcal{H}d$ with respect to some other root, which we label, again WLOG, z_2. We will establish a constrained polynomial optimization program with variables in $\mathbb{R}^\eta$ where $\eta = 6n - 8$.

Similar to our Notation 3.2, we define semialgebraic sets G_u, G_s and G_d as follows.

4.1.1 Notation

We introduce real variables. For degree $n = 2$, let

$$\begin{aligned} x_1 &:= \mathcal{R}[z_2] \\ x_2 &:= \mathcal{I}[z_2] \\ x_3 &:= \mathcal{R}[w_1] \\ x_4 &:= \mathcal{I}[w_1]. \end{aligned}$$

For degree $n \geq 3$, let

$$\begin{aligned} x_1 &:= \mathcal{R}[z_2] \\ x_2 &:= \mathcal{I}[z_2] \\ x_3 &:= \mathcal{R}[z_3] \\ x_4 &:= \mathcal{I}[z_4] \\ &\vdots \\ x_{2n-3} &:= \mathcal{R}[z_n] \\ x_{2n-2} &:= \mathcal{I}[z_n] \\ x_{2n-1} &:= \mathcal{R}[w_1] \\ x_{2n} &:= \mathcal{I}[w_1] \end{aligned}$$

$$\vdots$$

$$x_{4n-5} := \mathcal{R}[w_{n-1}]$$
$$x_{4n-4} := \mathcal{R}[w_{n-1}]$$

and the remaining variables $\{x_{4n-3}, \ldots, x_{6n-8}\}$ are used to compute $u\mathcal{H}d$.

4.1.2 Roots Contained within unit disk

For $1 \leq \ell \leq n-1$, label the following inequality constraints.

$$g_\ell := 1 - x_1^2 - x_2^2 \geq 0$$

Let $G_U = \{g_\ell : 1 \leq \ell \leq n-1\}$

4.1.2.1 Implicitly define relationship between roots and critical points

Recall that

$$Z := \{z_1, \ldots, z_n\} = \{1 + 0i, x_1 + x_2 i, \ldots, x_{2n-3} + x_{2n-2} i\}$$

and

$$W := \{x_{2n-1} + x_{2n} i, \ldots, x_{4n-5} + x_{4n-4} i\}$$

We parameterize Equation 2.7.

4.1.2.2 For Gloptipoly.

For n=2,

$$g_2 := x_1 + 1 - 2x_3 = 0$$
$$g_3 = x_2 - 2x_3 = 0$$

For $n \geq 3$, with $0 \leq \ell \leq n-2$,

$$
\begin{aligned}
g_{n+2\ell} &:= \mathcal{R}[\ell\sigma_{(n-\ell)}(Z) - n\sigma_{(n-\ell)}(W)] = 0 \\
g_{n+2\ell+1} &:= \mathcal{I}[\ell\sigma_{(n-\ell)}(Z) - n\sigma_{(n-\ell)}(W)] = 0
\end{aligned}
$$

Let $G_S := \{g_\ell : n \leq \ell \leq 3n-3\}$.

4.1.2.3 For YALMIP

For $n = 2$,

$$
\begin{aligned}
\gamma_2 &:= x_1 + 1 - 2x_3 \geq 0 \\
\gamma_3 &:= -\gamma_2 \geq 0 \\
\gamma_4 &:= x_2 - 2x_4 \geq 0 \\
\gamma_5 &:= -\gamma_4 \geq 0
\end{aligned}
$$

For $n \geq 3$, with $0 \leq \ell \leq n-2$,

$$
\begin{aligned}
\gamma_{n+4\ell} &:= \mathcal{R}[\ell\sigma_{(n-\ell)}(Z) - n\sigma_{(n-\ell)}(W)] \geq 0 \\
\gamma_{n+4\ell+1} &:= -\gamma_{n+4\ell} \geq 0 \\
\gamma_{n+4\ell+2} &:= \mathcal{I}[\ell\sigma_{(n-\ell)}(Z) - n\sigma_{(n-\ell)}(W)] \geq 0 \\
\gamma_{n+4\ell+3} &:= -\gamma_{n+4\ell+2} \geq 0
\end{aligned}
$$

Remark 4.1. . For $0 \leq r \leq 2(n-1)$:

$$\{\gamma_{n+4r}, \gamma_{n+4r+1}\} \text{ corresponds to } g_{n+2r}$$

and

$$\{\gamma_{n+4r+2}, \gamma_{n+4r+3}\} \text{ corresponds to } g_{n+2r+1}$$

Let $\Gamma_S := \{\gamma_n, \dots, \gamma_{5n-5}\}$.

4.1.3 $u\mathcal{H}d$

We maximize with $u\mathcal{H}d$ with respect to z_2. I.e., z_2 will be one of roots with maximal distance from any critical point. The distance of z_2 to closest critical point is

$$\min_{w \in W} |z_2 - w|$$

Furthermore, because we interfaced with optimization programs which by default minimize, our optimization was of form:

$$\min_{F} 1 - \Delta(F)$$

For $n = 2$ we directly describe this distance as:

$$g_4 := 1 - \left[(x_1 - x_3)^2 + (x_2 - x_4)^2\right].$$

4.1.3.1 For Gloptipoly.

For $n = 3$, we apply Equation 4.2

$$\begin{aligned} g_7 &:= [((x_1 - x_5)^2 + (x_2 - x_6)^2) - ((x_1 - x_7)^2 + (x_2 - x_8)^2)]^2 - x_9^2 = 0 \\ g_8 &:= x_9 \geq 0 \\ g_9 &:= ((x_1 - x_5)^2 + (x_2 - x_6)^2) + ((x_1 - x_7)^2 + (x_2 - x_8)^2) - x_9 - 2x_10 = 0 \end{aligned}$$

and $1 - x_10$ will be our objective function.

For $n \geq 4$, the first triplet is similar to the above and then we iteratively apply 4.2 as follows. For $0 \leq \ell \leq n - 3$,

$$\begin{aligned} g_{(3n-2)+3\ell} &:= \left[((x_1 - x_{2n+2\ell-1})^2 + (x_2 - x_{2n+2\ell})^2) - ((x_1 - x_{2n+2\ell+1})^2 + (x_2 - x_{2n+2\ell+2})^2)\right]^2 - x_{4n-4+2\ell+1} = 0 \\ g_{(3n-2)+3\ell+1} &:= x_{4(n-1)+2\ell+1} \geq 0 \\ g_{(3n-2)+3\ell+2} &= (x_1 - x_{2n+2\ell-1})^2 + (x_2 - x_{2n+2\ell})^2 + (x_2 + x_{2n+2\ell+1})^2 + (x_2 + x_{2n+2\ell+2})^2 - x_{4(n-1)+2\ell+1} - 2x_{4(n-1)+2\ell+2} = 0 \end{aligned}$$

and $1 - x_{6n-8}$ will be our objective function.

When $n = 2$ let $G_U = \emptyset$. Otherwise, let

$$G_D := \{g_\ell : 3n-2 \le \ell \le 6n-9$$

and we check $|G_U| = 3(n-2)$. Partition G_U into equations and inequalities as follows.

$$G_{D,e} := \{g_{3n-2+3\ell}, g_{3n-2+3\ell+2}\}_{0\le\ell\le n-3}$$

$$G_{D,ineq.} := G_U \backslash G_{U,e}$$

4.1.3.2 For YALMIP.

Similarly, for $n = 3$,

$$\begin{aligned}
&\gamma_{11} := \left[((x_1-x_5)^2 + (x_2-x_6)^2) - ((x_1-x_7)^2 + (x_2-x_8)^2)\right]^2 - x_9^2 \ge 0\\
&\gamma_{12} := -\gamma_{11} \ge 0\\
&\gamma_{13} := x_9 \ge 0\\
&\gamma_{14} := (x_1-x_5)^2 + (x_2-x_7)^2 + (x_1-x_7)^2 + (x_2-x_8)^2 - x_9 - 2x_{10} \ge 0\\
&\gamma_{15} := -\gamma_{14} \ge 0
\end{aligned}$$

For $n \ge 4$, the first pentad is similar to that of $n = 3$, and then the other pentads are iterative, as follows. For $0 \le \ell \le n-3$,

$$\begin{aligned}
\gamma_{5n-4+5\ell} &:= \left[((x_1-x_{2n+2\ell-1})^2 + (x_2-x_{2n+2\ell})^2) - ((x_1-x_{2n+2\ell+1})^2 + (x_2-x_{2n+2\ell+2})^2)\right]^2 - x_{4(n-1)+2\ell+1} \ge 0\\
\gamma_{5n-4+5\ell+1} &:= -\gamma_{5n-4+5\ell} \ge 0\\
\gamma_{5n-4+5\ell+2} &:= x_{4(n-1)+2\ell+1} \ge 0\\
\gamma_{5n-4+5\ell+3} &:= (x_1-x_{2n+2\ell-1})^2 + (x_2-x_{2n+2\ell})^2 + (x_1-x_{2n+2\ell+1})^2 + (x_2-x_{2n+2\ell+2})^2 - x_{4(n-1)+2\ell+1} - 2x_{4(n-1)+2\ell+2} \ge 0\\
\gamma_{5n-4+5\ell+4} &:= -\gamma_{(5n-4)+5\ell+3} \ge 0
\end{aligned}$$

Collect these constraints as follows. When $n = 2$, $\Gamma_D := \emptyset$. For $n \ge 3$, $\Gamma_D = \{\gamma_{5n-4}, \ldots, \gamma_{10n-15}\}$ and verify $|\Gamma_D| = 5(n-2)$.

Remark 4.2. For $0 \le r \le n-4$,

$$\{\gamma_{(5n-4)+5r}, \gamma_{(5n-4)+5r+1}\} \text{ corresponds to } g_{3n-2+3r}$$

$$\gamma_{(5n-4)+5r+1} \text{ corresponds to } g_{(3n-2)+3r+1}$$

and

$$\{\gamma_{(5n-4)+5r+3}, \gamma_{(5n-4)+5r+4}\} \text{ corresponds to } g_{(3n-2)+3r+2}$$

4.1.3.3 Constraints define a semialgebraic set, our feasible region.

Let $G := G_U \sqcup G_{D,ineq}$, $H := G_S \sqcup G_{D,eq.}$, and $\Gamma := \Gamma_U \sqcup \Gamma_S \sqcup \Gamma_D$. Define the semialgebraic set K as follows

$$\begin{aligned} K &= \{\vec{x} \in \mathbb{R}^\eta | \forall g \in G, g(\vec{x}) \geq 0 \bigwedge \forall h \in H, h(\vec{x}) = 0\} \\ &= \{\vec{x} \in \mathbb{R}^\eta | \forall \gamma \in \Gamma, \gamma(\vec{x}) \geq 0\} \end{aligned}$$

4.1.3.4 Objective Function.

Define

$$g_0 = \begin{cases} 1 - (x_1 - x_3)^2 - (x_2 - x_4)^2 & \text{when } n = 2 \\ 1 - x_{10} & \text{when } n = 3 \\ 1 - x_{6n-8} & \text{when } n \geq 4 \end{cases}$$

Conjecture 2.3 is affirmed when a certificate [40, Theorem 2.12(a) Nichtnegativstellensatz] for g_0 on the semialgebraic set K is provided.

In other words, Conjecture 2.3 is equivalent to

$$0 = \inf_{\vec{x} \in K} g_0(\vec{x}) \tag{4.3}$$

4.1.3.5 Positivity Certificates

In the case of Gloptipoly our search space for certificates is a quadratic module

$$\sigma_0 = g_0 - \gamma - \sum_{1 \leq \ell \leq 6n-9} \sigma_\ell g_\ell$$

where γ is a lower bound of g_0.

In the case of reporting results from YALMIP, post-process to match the above form. This involves re-pairing the two positive semidefinite matrices corresponding to one equality constraint. I.e., because YALMIP outputs

$$\sigma_0 = g_0 - \gamma - \sum_{1 \leq q \leq 10n-14} \tau_q g_q$$

then let $\sigma_q = \tau_{q_1} - \tau_{q_2}$ for those correspondences found in Remark 4.1 and Remark 4.2.

Explicitly repeated here: for $0 \leq r \leq (n-4)$

$$\begin{aligned}
\sigma_{(3n-2)+3r} &= \tau_{(5n-4)+5r} - \tau_{(5n-4)+5r+1} \\
\sigma_{(3n-2)+3r+2} &= \tau_{(5n-4)+5r+3} - \tau_{(5n-4)+5r+4}
\end{aligned}$$

and for $0 \leq s \leq (n-3)$

$$\begin{aligned}
\sigma_{n+2s} &= \tau_{(n+4s} - \tau_{(5n-4)+5r+1} \\
\sigma_{n+2s+1} &= \tau_{(5n-4)+5r+3} - \tau_{(5n-4)+5r+4}
\end{aligned}$$

Our Gloptipoly results, a selection of which is included in Appendix A.3, encouraged us to seek closed forms of our positivity certificates.

We then used YALMIP to provide explicit positivity certificates, up to machine precision (aka double). See for example, Appendix A.2. Our hope was to find a pattern that might be found in explicitly positivity certificates, and to generally across each $SCFD_n$. Our numerical experiments of 2011-2012 were largely. done in two different contexts one with "all zeros free" 2.2 and one with "one zero fixed at 1" 2.17.

All of these experiments involved "bootstrapping" $u\mathcal{H}d$ or its equivalent. Above, we provide an implementation in which we applied only apply 4.2. In Appendix A.4, we provide an implementation which iteratively applies both 4.2 and 4.1 in order to fully boostrap $u\mathcal{H}d$. We wrote several scripts, which we eventually translated to Java for purposes of unit testing with JUnit, to generate the MATLAB scripts which would invoke either Gloptipoly or YALMIP. The interfaces for each of Gloptipoly and YALMIP were slightly different, so we modified accordingly our code.

Let $\mathbb{C}[z]_n$ is the set of univariate complex polynomials of degree n or less. We focus on applying optimization techniques to Conjecture 2.3.

4.2 Results and Concluding Remarks

Our supporting numerical evidence was found based upon methods in the intersection of moment theory, positive polynomials, semidefinite programming (SDP), LMI techniques, convex programming, and optimization. In this chapter, we presented a selection of our experiments. Specifically, scripts for Gloptipoly3 with SeDuMi. In addition, scripts for YALMIP. Thus, we have expressed Conjecture 2.2 as an input to optimization programs including Gloptipoly [31][37][30][40], SOSTools [57][56][55][53][54], YALMIP [31] [44][45][43], and with various back-end solvers [50]: including SDPT [75], SparsePOP , Sedumi [73], CVXOPT [2][14], CSDP [13], SDPA [80]. Numerous experiments were performed circa 2011-2012 on the clusters of CNSI and the Center for Scientific Computing at the University of California, Santa Barbara. Occassionally, between 2012-2015 in our author's spare time, we returned to these experiments, reproduced them, and expanded upon them.

Remark 4.3. In general, it was easier to interpret our results when we focused our inputs to characterize Conjecture 2.17. We also performed the experiments in the context of our fully symmetric Conjecture 2.2.

Remark 4.4. For degrees higher than have yet to be proved (i.e. $9 \leq n \leq 14$), the semidefinite programming based results provided similar output as to those for degrees $3 \leq n \leq 8$.

Remark 4.5. In addition, it is our awareness that the positivity certificates sought by both Gloptipoly and YALMIP seek certificates of form [40, Theorem 2.14 Putinar's Positivstellensatz] rather than [40, Theorem 2.12 Stengle's Nullstellensatz]. The tradeoff for better computational complexity is lack of search throughout the complete space of certificates.

Remark 4.6. Numerical evidence provides certificates only up to machine error.[5] Via the method of semidefinite optimization, we found difficulty with obtaining closed forms by parsing the numerical certificates (i.e. matrices translated back into sums-of-squares polynomials) as the coefficients of the polynomials were double float values. Thus, we find that while the numerical experiments in sums-of-squares programming provided much numerical evidence for the conjecture, these experiments did not provide proof. In redemption, had there been a counterexample found by these techniques beyond machine, it would have provided evidence refuting our conjecture.

[5]Only in the case of $n = 2$ did we achieve both of the following: 'numerical certification of global optimality' within most strict tolerance on machine error *and* all solutions found were feasible for every time the script was run on a machine. Please refer to table found in Figure 4.1 of Appendix A.4 for some details. Since we are on the subject of error, by comparison to other root finding algorithms, the use toward rootfinding (numerical computation of roots) of gloptipoly, yalmip or sostools in the manner of our scripts may provide impressive performance, in particular accuracy.

Chapter 5

Dynamics

This chapter contains a selection of material found in Appendix A.4[1].

5.1 Sending One Root Out

With Z as our root set, for this section, suppose that we fix all the roots besides one. In this case, if z_n is a multiple root, we intend only for one multiplicity to be replaced. In other words, we are replacing only one root. Without loss of generality (in other words, up to relabelling), let this free root be z_n. Furthermore, let its polar decomposition be $z_n = R_n e^{i\theta_n}$.

Now, consider one real parameter

$$R \in [R_n, \infty) \subset \mathbb{R}$$

and the family of roots parameterized by replacing z_n with $Re^{i\theta_n}$.

Notation. Let $z_R = Re^{i\theta n}$,

$$Z_R = \{z_1, \ldots, z_{n-1}, z_R\} = \{z_R\} \cup (Z \backslash \{z_n\}),$$

and

[1] c.f. "OZOAB"

$$Z_{-1} = \{z_1, \ldots, z_{n-1}\} = Z \backslash \{z_n\}$$

Further, maintain the assumption that $Z_{-1} \subset \mathbb{D}$.

Lemma 5.1

For sufficiently large R, there is an M for which there exists a unique critical point outside of a circle centered at the origin of radius M.

Proof. We will perform a standard, well-known application of Rouche's theorem[2] Consider 2.1, rearrange terms and take moduli of each, let M be the modulus of z, and then consider Equation 2.9 while expressing coefficients. To obtain, with $\overset{?}{>}$ denoting the inequality we are to show:

$$|(n-1)\sigma_1(Z_R\})|M^{n-2} \overset{?}{>} nM^{n-1} + \sum_{\ell=2}^{n-1} |(n-\ell)\sigma_\ell(Z_R)|M^{n-\ell-1} \tag{5.1}$$

so that we may conclude by Rouche's theorem that there are $n-2$ critical points in the circle $|z| = M$.

If $n = 2$, the result follows for $M = 1$ and $R > 3$. Assume $n \geq 3$, if we find for R large then our result will follow.

First, consider the coefficient on the left side of Inequality 5.1

$$\begin{aligned}
|(n-1)\sigma_1(Z_R)| &= |(n-1)(z_1 + \ldots + z_{n-1} + z_R)| \\
&\text{By the triangle inequality:} \\
&\geq (n-1)(R - |z_1 + \ldots + z_{n-1}|) \\
&\text{By the unit disc constraint for the fixed n-1 roots} \\
&\geq (n-1)(R-(n-1))
\end{aligned} \tag{5.2}$$

Second, consider the right side of Inequality 5.1

[2] Special thanks to David Damanik [23] and Nikolai Makarov [46] for instruction and assignments of these materials circa 2003. c.f. [Section 10.2 Applications of the Residue Theorem: 10.10 Rouche's Theorem (pg.123), Example (pg. 124), and Problem 6 parts c and d (pg. 128)][4] and [Section 5.2 The Argument Principle: Corollary (pg. 153) and Exercises 1 and 2 (pg. 154)][1]

$$nM^{n-1} + \sum_{\ell=2}^{n-1} \left|(n-\ell)\sigma_\ell(Z_R)\right| M^{n-\ell-1}$$

Which is equivalent, by Equation 2.5) to

$$nM^{n-1} + \sum_{\ell=2}^{n-1} |(n-\ell)(z_R\sigma_{\ell-1}(Z_{-1}) + \sigma_\ell(Z_{-1}))|M^{n-\ell-1}. \tag{5.3}$$

Now, the modulus of any product of elements in Z_- is 1 or less because $Z_- \subset \mathbb{D}$. Furthermore, there are respectively $\binom{n-1}{\ell-1}$ and $\binom{n-1}{\ell}$ products in $\sigma_{\ell-1}(Z_-)$ and $\sigma_\ell(Z_-)$. Thus, 5.3 is less than

$$nM^{n-1} + \sum_{\ell=2}^{n-1}(n-\ell)\left(R\binom{n-1}{\ell-1} + \binom{n-1}{\ell}\right) M^{n-\ell-1} \tag{5.4}$$

Now, we pick $M = R/2$. Let $O(\cdot)$ denote Big-O notation so that 5.4 becomes

$$n\left(\frac{R}{2}\right)^{n-1} + O(R^{n-2})$$

Thus, Inequality 5.1, follows when R is such that

$$(n-1)(R-(n-1))\left(\frac{R}{2}\right)^{n-2} > n\left(\frac{R}{2}\right)^{n-1} + O(R^{n-2}) \tag{5.5}$$

which is equivalent, by collecting powers of $\frac{R}{2}$, to

$$(2(n-1)-n)\left(\frac{R}{2}\right)^{n-1} > 2(n-1)^2\left(\frac{R}{2}\right)^{n-2} + O(R^{n-2}) = O(R^{n-2}). \tag{5.6}$$

Therefore, there is a sufficiently large R such that Inequality 5.1 follows. $\square$

Lemma 5.2

For sufficiently large R, there will be, including multiplicty, $n-2$ critical points contained within the circle $|z| = M = 2n$.

Proof. Similar to the proof of 5.1: consider Equation 2.9 and collecting terms of appropriate powers proceed towards an application of Rouche's theorem. This time, set $M = 2n$

and collect R on the left side to obtain:

$$R\left((n-1)M^{n-2}-\sum_{\ell=2}^{n-1}(n-\ell)\binom{n-1}{\ell-1}M^{n-\ell-1}\right)$$

$$\geq$$

$$(n-1)^2M^{n-2}+nM^{n-1}+\sum_{\ell=2}^{n-1}\binom{n-1}{\ell}M^{n-\ell-1}$$

for sufficiently large R. $\square$

Lemma 5.3

For sufficiently large R there will be, including multiplicity, $n-2$ critical points contained within an epsilon distance from the unit circle.

Proof. The bounds in the proof Lemma 5.1 and Lemma 5.2 are poor indeed, but they allow us to make the assumption that for $n-2$ critical points, $|w| \leq 2n$, when R is large. If $w = z_k$ for $1 \leq k \leq n-1$ then we are done.

Recall from Proposition 2.15, that

$$w=\sum_{k=1}^{n}\lambda_k z_k=\lambda_n Re^{i\theta_n}+\sum_{k=1}^{n-1}\lambda_k z_k$$

and further recall Equation 2.16, our definition of λ_n:

$$\begin{aligned} 0 \leq \lambda_n &= \frac{1/|w-z_n|^2}{\sum_{\ell=1}^{n} 1/|w-z_\ell|^2} \\ &\text{By removing one positive term in the denominator} \\ &\leq \frac{1/|w-z_n|^2}{\sum_{\ell=1}^{n-1} 1/|w-z_\ell|^2} \\ &= \frac{A}{|w-z_n|^2} \end{aligned} \tag{5.7}$$

where

$$A = \frac{1}{\sum_{\ell=1}^{n-1} 1/|w - z_\ell|^2}$$

.

Because $|w| \leq 2n$,

$$\begin{aligned} &|w - z_\ell| \leq 2n + 1 \\ \Rightarrow &\frac{1}{(2n+1)^2} \leq \frac{1}{|w - z_\ell|^2} \\ \Rightarrow &\frac{n-1}{(2n+1)^2} \leq \sum_{k=1}^{n-1} \frac{1}{|w - z_\ell|^2} \\ \Rightarrow &A \leq \frac{(2n+1)^2}{n-1} \end{aligned}$$

Furthermore, $|w| \leq 2n$ also implies

$$|w - z_n|^2 = |w - Re^{i\theta}|^2 \geq (R - 2n)^2$$

$$\Rightarrow$$

$$\lambda_n \leq \frac{A}{(R-2n)^2} \leq \frac{(2n+1)^2}{(n-1)(R-2n)^2} \tag{5.8}$$

Now, $\{z_k\}_{k=1}^{n-1} \in \mathbb{D}$, so

$$\begin{aligned} |w| &\leq \lambda_n R + \sum_{k=1}^{n-1} \lambda_k = \lambda_n R + (1 - \lambda_n) \\ &= 1 + \lambda_n (R - 1) \\ &\leq 1 + \frac{(2n+1)^2}{(n-1)} \frac{R-1}{(R-2n)^2} \end{aligned}$$

Thus, for large R, $|w| \leq 1 + \epsilon$. $\square$

Corollary 5.4

Furthermore,

$$\lim_{R\to\infty} \lambda_n z_n = \lim_{R\to\infty} A\frac{R}{(R-2n)^2} = 0$$

Therefore, as $\mathbb{R} \to \infty$,

$$w \sim \sum_{k=1}^{n-1} \lambda_k z_k$$

for the $n-2$ critical points which remain near the unit disk.

Remark 5.1. In other words, asymptotically, the critical points that get left behind eventually behave as if there are only the roots Z_-.

Remark 5.2. Roughly speaking, as we let R increase, we are 'sending one root out.' We notice that one critical point 'comes for the ride.' Without resorting to the full rigor of the previous lemmas, consider the asymptotics of the critical point that travels along with z_n. Suppose $|w| >> 1$ and $(w - z_k) \sim w$ for $k \neq n$. Equation 2.14 becomes

$$0 \sim \frac{1}{w - z_n} + \frac{n-1}{w}$$

$$w \sim (n-1)(z_n - w)$$

$$w \sim \frac{n-1}{n} z_n$$

which agrees well with numerical experiments.

We summarize the previous two remarks in the following Lemma.

Lemma 5.5

For R large, there will be one critical point at

$$w \sim \frac{n-1}{n} z_n$$

and $n-2$ critical points at

$$w \sim \sum_{k=1}^{n-1} \lambda_k z_k$$

Proof. Formally, Lemma 5.3 and Corollary 2.13 confirm. □

Consider the following examples

Example 5.6. Let $f(z) = (z - R)(z - 1)^{n-1}$.

$$f'(z) = (z-1)^{n-1} + (z-R)(n-1)(z-1)^{n-2} = (z-1)^{n-2}((z-1) + (n-1)(z-R))$$

There will be a critical point at 1 with multiplicity $n - 2$ and one critical point at

$$w = \frac{(n-1)R + 1}{n}$$

Example 5.7. Let $f(z) = (z - R)(z + 1)^{n-1}$.

$$f'(z) = (z+1)^{n-1} + (z-R)(n-1)(z+1)^{n-2} = (z+1)^{n-2}((z+1) + (n-1)(z-R))$$

There will be a critical point at -1 with multiplicity $n - 2$ and one critical point at

$$w = \frac{(n-1)R - 1}{n}$$

Example 5.8. Let $f(z) = (z - R)(z^2 + 1)$. Then

$$f'(z) = (z^2 + 1) + (z - R)(2z) = 3z^2 - 2Rz + 1$$

$$w = \frac{2R \pm \sqrt{4R^2 - 12}}{6}$$

Example 5.9. Let $0 < t << 1$. I.e., let t be small and positive. Let

$$f(z) = (z - e^{it})(z - e^{-it})(z - R)$$

The for some values of small $R > 1$, no critical points will be within the unit circle.

$$f'(z) = (z-e^{-it})(z-R)+(z-e^{it})(z-R)+(z-e^{it})(z-e^{-it}) = 3z^2-2(e^{-it}+e^{it}+R)z+(R(e^{-it}+e^{it})+1)$$

$$w = \frac{2(2\cos(t) + R) \pm \sqrt{(2\cos(t) + R)^2 - 12(2R\cos(t) + 1)}}{6}$$

Remark 5.3. As an imaginative aside, we ask the following. Can we show that $\Delta(Z_\epsilon) \leq 1$ for any perturbation of the roots such that each is contained within the unit disk and is within a uniform distance from an original roots of unity? I.e. $|Z_\epsilon| = n$ and

$$\forall z \in Z_\epsilon : \left(|z| \leq 1 \text{ and } |z_k - e^{i\frac{2\pi}{n}}| < \epsilon \right)$$

Chapter 6

Conclusion

Thus far, we have only found, in closed form, certificates 3.5 for $SCFD$ in the case of $n = 2$.

All numerical evidence, including scripts written in Matlab and Octave (e.g. Gloptipoly, SeDuMi, YALMIP), Singular, Java (c.f. Scala), GAMS, Python and SageMath, provided further evidence toward Conjecture 0.1 being true. Various scripts for Matlab, GAMs, Singular, and other programs can be obtained by modifying (insert reference to Java electronic resources) or the ideal generation paragraphs of (insert reference to python paragraphs or python electronic resources).

In other words, for degrees $n > 2$, even when we allowed for larger relaxation orders or greater number than the default bound on iterations: the program either did not terminate or some subset of candidate certificates were infeasible, which is understandable given the apparent tightness of 1 being our $u\mathcal{H}d$ bound. Whenever we terminated[1] the ongoing processes of programs in the fully symmetric case (c.f. Remark 4.3) we found[2] rotations of the roots of unity. Whenever we terminated the programs in the context of

[1]This is a computer science/software engineer/programmer/numerics term which basically means we stop an iterative process prematurely. In the context of the mathematics involved, we stopped our semidefinite programming routine before the duality gap became smaller than our pre-specified bound because the process was taking too long.

[2]In the 'queue of best polynomials' meaning those solutions, found by the routine thus far (prior to termination) that were feasible and had largest $u\mathcal{H}d$

Section 4.1, we found that the feasible solutions were (up to machine error) the canonical roots of unity (i.e. $\{e^{i\pi k/n} : 0 \leq k \leq n-1\}$). Numerical experiments were performed across different computing environments and we simply do not have the resources to report here upon all of them. We provide output of experiments for $n = 2$ and $n = 3$. Results for higher degrees can be reproduced with the information (including source code) provided in this dissertation.

It is possible that the relaxations for $n \geq 3$ as we coded them with the interfaces available at that time, some of which involve a hierarchy of sums-of-squares *only* in a Quadratic Module rather than its corresponding Preordering, were insufficient[3]. Also, our optimization searches did not include an additional sum-of-squares denominator which would make the search for the certificate in the space of rational sums of squares [7]. For my own limitations, I did not prove that the certificates can only be rational sums of squares with non-trivial denominator. To be clear, about what is meant by "non-trivial denominator," we provide the following version of Stengle's Nichtnegativestellensatz as an example[4], which itself closely follows [40, Section 2.4 Representation Theorems: Multivariate Case, Theorem 2.12(a) Stengle's Nichtnegativstellensatz, pg. 28]. Let $\mathbb{K}$ generated by a finite set $\{f_j\} \subset k[x]$ where k is a real closed field. For $f \in k[x]$, the following are equivalent:

- $f \geq 0$ on $\mathbb{K}$
- $fg = f^{2m} + h$

where $m \in \mathbb{N}$[5] For example, if we have the hypotheses of Putinar's Positivstellenatz, then our search-space for certificates is smaller (we need search only in a quadratic module rather than a preorder) and $g = 1$. I did not prove that the relevant Quadratic Modules

[3]because for any particular non-trivial set of generators, their preordering is a strict superset of their quadratic module

[4]Though we can communicate this in the context of other Positivstellensatz

[5]There is a correspondence between this m and the m found in Chapter 3 which is also the m of the second paragraph of the code presented in Appendix is that $g \neq 1$ with g and$h$ in a preordering.

for the semialgebraic sets which we use to describe Sendov's conjecture are Archimedean. Some numerical evidence suggests that it might not be the case for $n > 2$, but I did not prove that our semialgebraic sets are not Archimedean, either.

In 2012, we introduced additional novel approaches toward proof of Conjecture 0.1: one involving dynamics and analysis[6] and another involving topology and calculus [7]. In the parlance of our advisor (communicated during May 2012 defense), we explored pullbacks for our canonical differential operator which at that time, we called Acceptable Constants Of Integration. However, work done[8] since 2012 has shown that neither would result in a concise proof and it's unlikely a proof will come from these lines of inquiry. In the case of the dynamics approach, when we follow the critical point which comes with the root back (i.e., bring $R \to R_n^+$) it can interact and potentially coincide with other critical points. This fact also sheds light on why other approaches in Real Algebraic Geometry, such as the Cylindrical Algebraic Decomposition (CAD), are indeed messy in application because our configuration space[9] is tricky. We did perform CAD, by hand, for $n = 2$ and performed part of the algorithm for $n = 3$, prior to reaching this conclusion.

We make use of the Fundamental Theorem of Algebra, which can be expressed as an application of Liouville's theorem, which can be expressed as an application of the Cauchy Integral Formula. In particular, some proofs of Liouville's theorem are not constructive. Thus, it is my intuition that when we express Sendov's conjecture we are expressing a statement in higher-order[10] logic[11] because we can express Sendov's conjecture with universal quantifier over polynomials and then other quantifiers over roots[12]. We mention this because there are deep waters here. Take care, dear reader.

We hope for opportunities for further research. In particular, with respect to this

[6] c.f. Chapter 5
[7] c.f ACOI in Appendix A.4
[8] some by our present author in this document and other excellent work [59]
[9] moduli spaces. Voronai partitions.
[10] than first-order
[11] in some Logical Models which harmonize with classical Complex Analysis.
[12] of both original function and roots

work and posterity, our best bet is to continue running our SageScripts to search for the degree 3 real radical ideal membership and modifying them consideration of choosing implementations of Algorithm 3.1 which make the search feasible. Finding a few more lower order certificates will help guide a general approach in hope for finding a closed form.

Appendix A

Appendix

A.1 Validation Of Naive Implementation of Algorithm 3.1

Remark A.1. The following notebook is a "validation version" adapted from a more general implementation. Its source can be found in our Supplementary Material [69]. The comment in this notebook regarding row-reduction does not apply. The paragraphs of which correspond to the step, "solve the systems of equations," of Algorithm 3.1 found in other notebooks of were replaced with a unit test. Here, our unit test is a verification Equation 3.4.[1]

[1] That's the nature of comments and paragraph format, and although I'm slightly embarrassed by the long-winded names I chose in the code [especially as it requires display breaks] the code in notebook form is prototyping...and long-winded, sufficiently detailed names can sometimes avoid misinterpretation [and can also aid disambiguation when using grep].

ValidationOfNaiveImplementation

June 12, 2020

[1]:
```
#Goal: set up ipynb to search for real radical ideal membership
#i.e., an integer m, an sos, and an an ideal member such that
#-t_1^(2m)
# is expressed as a sum of squares plus an ideal member
#note that the above is only one monomial.
# I.e. all zero coefficients except for one monomial!
#we parameterize the s.o.s.
#and parameterize ideal memberhsip
#[*]and then row-reduce and see if there are any non-zero solutions
#...which there should be...
#because we know our conjecture is true for low degree
#[*]...we hope to find 'nice' integral solutions
```

[2]:
```
#Hyperparameters for this script. All integers.

#Our conjecture for fixed degree
n=2
assert(n>=2)

#This is our guess for the smallest m
#which will describe t_1 as a real radical member
m=1
assert(m>=1)

#Number of sums of squares (hereafter, SOS)
pythagoras_bound=6
assert(pythagoras_bound>=5)

#Degrees of polynomials to be parameterized and squared (in SOS)
sos_parameterization_degree=1
assert(sos_parameterization_degree>=1)

#Max degree of parameterized ideal membership
maximal_degree_of_monomials_to_collect_ideal_membership=4
assert(maximal_degree_of_monomials_to_collect_ideal_membership>2)
```

[3]:
```
def generate_variables(character,number):
    return [character+'%s'%x for x in range(1,number+1)]

#real parts of roots
A=generate_variables('a',n)
#imaginary parts of roots
B=generate_variables('b',n)
#real parts of critical points
C=generate_variables('c',n-1)
#imaginary parts of critical points
D=generate_variables('d',n-1)
#slack variables for unit disk constraint on roots
S=generate_variables('s',n)
#slack variables for WLOG distance between a chosen root
# and each critical point is at least one
T=generate_variables('t',n-1)

#Define subsets of variable labelings
Roots = A+B
CP=C+D
RootsAndCP=Roots+CP
Slack=S+T
Send=Roots+CP+Slack

#we define this function to tie together,
#conceptually, the functions which use it
def get_wiggle_room_from_poly(maximal_degree_of_monomials_to_collect, poly):
    return maximal_degree_of_monomials_to_collect-poly.degree()
def get_wiggle_room_from_poly_degree(maximal_degree_of_monomials_to_collect,␣
 ↪poly_degree):
    return maximal_degree_of_monomials_to_collect-poly_degree
from sage.combinat.integer_vector_weighted import iterator_fast
def␣
 ↪get_list_of_multiindex_of_degrees_of_monomials_per_wiggle_room(wiggle_degree,␣
 ↪variable_set):
    #print("\t Wiggle degree: {}".format(wiggle_degree))
    return list(iterator_fast(wiggle_degree,[1]*len(variable_set)))
def get_number_of_monomials_per_wiggle_room(wiggle_degree,variable_set):
    degs= len(␣
 ↪get_list_of_multiindex_of_degrees_of_monomials_per_wiggle_room(wiggle_degree,␣
 ↪variable_set))
    return degs

#compute number of necessary parameters for ideal membership parameterization
def compute_number_of_monomials_to_describe_ideal_membership(n,␣
 ↪maximal_degree_of_monomials_to_collect_ideal_membership):
```

2

```
    wiggle =␣
↪get_wiggle_room_from_poly_degree(maximal_degree_of_monomials_to_collect_ideal_membership,2)
    number_of_new_parameters_per_degree_2_poly =␣
↪get_number_of_monomials_per_wiggle_room(wiggle, RootsAndCP)
    #we know that there will be n unit_circle_constraints (each degree 2)
    #and n-1 distance_constraints (also each degree 2)
    retval = (2*n-1)*number_of_new_parameters_per_degree_2_poly
    for sym_pol_degree in range(1,n):
        #two symmetric polynomials constraints per degree
        #(real and imaginary parts)
        wiggle = get_wiggle_room_from_poly_degree(\
                    maximal_degree_of_monomials_to_collect_ideal_membership,␣
↪sym_pol_degree)
        retval += 2*get_number_of_monomials_per_wiggle_room(wiggle, RootsAndCP)
    return retval

number_of_parameters_for_ideal_membership =␣
↪compute_number_of_monomials_to_describe_ideal_membership(n,
                                                         ␣
↪maximal_degree_of_monomials_to_collect_ideal_membership)

Param=generate_variables('h',number_of_parameters_for_ideal_membership)

#compute number of necessary parameters for sos parametrization
maximal_degree_of_monomials_to_collect_sos = 2* sos_parameterization_degree
from scipy.special import comb as comb_choose
def get_number_of_monomials_for_sos(number_of_variables,max_deg):
    _mon_deg = max_deg/2
    return comb_choose(number_of_variables+_mon_deg,_mon_deg)
num_monomials_for_sos = get_number_of_monomials_for_sos(
    len(Send),maximal_degree_of_monomials_to_collect_sos)
number_of_parameters_for_sos = pythagoras_bound*int(num_monomials_for_sos)
Param_SOS = generate_variables('g',number_of_parameters_for_sos)

Total = Send+Param+Param_SOS

#R=PolynomialRing(QQ,Total,order='degrevlex')
R=PolynomialRing(QQ,Total,order='lex')
R.inject_variables()

#now redefine our subsets of variable labelings
# as references to the variable objects
A=R.gens()[0:n]
B=R.gens()[n:2*n]
C=R.gens()[2*n:3*n-1]
D=R.gens()[3*n-1:4*n-2]
S=R.gens()[4*n-2:5*n-2]
```

```
T=R.gens()[5*n-2:6*n-3]
Param=R.gens()[6*n-3:6*n-3+number_of_parameters_for_ideal_membership]
Param_SOS = R.gens()[6*n-3+number_of_parameters_for_ideal_membership:]
Roots = A+B
CP=C+D
RootsAndCP=Roots+CP
Slack=S+T
Send =Roots+CP+Slack

def get_unit_circle_constraint(index):
    return A[index]^2+B[index]^2+S[index]^2-1

unit_circle_constraints = [get_unit_circle_constraint(index) for index in␣
 ↪range(n)]
print(unit_circle_constraints)
def get_critical_point_distance_from_first_root(index):
    return (A[0]-C[index])^2+(B[0]-D[index])^2-T[index]^2-1

distance_constraints = [
    get_critical_point_distance_from_first_root(index) for index in range(n-1)
]
print(distance_constraints)
```

```
Defining a1, a2, b1, b2, c1, d1, s1, s2, t1, h1, h2, h3, h4, h5, h6, h7, h8, h9,
h10, h11, h12, h13, h14, h15, h16, h17, h18, h19, h20, h21, h22, h23, h24, h25,
h26, h27, h28, h29, h30, h31, h32, h33, h34, h35, h36, h37, h38, h39, h40, h41,
h42, h43, h44, h45, h46, h47, h48, h49, h50, h51, h52, h53, h54, h55, h56, h57,
h58, h59, h60, h61, h62, h63, h64, h65, h66, h67, h68, h69, h70, h71, h72, h73,
h74, h75, h76, h77, h78, h79, h80, h81, h82, h83, h84, h85, h86, h87, h88, h89,
h90, h91, h92, h93, h94, h95, h96, h97, h98, h99, h100, h101, h102, h103, h104,
h105, h106, h107, h108, h109, h110, h111, h112, h113, h114, h115, h116, h117,
h118, h119, h120, h121, h122, h123, h124, h125, h126, h127, h128, h129, h130,
h131, h132, h133, h134, h135, h136, h137, h138, h139, h140, h141, h142, h143,
h144, h145, h146, h147, h148, h149, h150, h151, h152, h153, h154, h155, h156,
h157, h158, h159, h160, h161, h162, h163, h164, h165, h166, h167, h168, h169,
h170, h171, h172, h173, h174, h175, g1, g2, g3, g4, g5, g6, g7, g8, g9, g10,
g11, g12, g13, g14, g15, g16, g17, g18, g19, g20, g21, g22, g23, g24, g25, g26,
g27, g28, g29, g30, g31, g32, g33, g34, g35, g36, g37, g38, g39, g40, g41, g42,
g43, g44, g45, g46, g47, g48, g49, g50, g51, g52, g53, g54, g55, g56, g57, g58,
g59, g60
[a1^2 + b1^2 + s1^2 - 1, a2^2 + b2^2 + s2^2 - 1]
[a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
```

[4]:

```
#Generate symmetric polynomial constraints
class Expression:
    def __init__(self, expression):
        self.expression = expression
```

4

```python
    def add(self, other):
        if self.expression==0:
            return other
        elif other.expression==0:
            return self
        else:
            return Expression(self.expression+other.expression)

    def subtract(self, other):
        return Expression(self.expression-other.expression)

    def mult(self, other):
        if (self.expression==0) or (other.expression==0):
            return Expression(0)
        elif self.expression==1:
            return other
        elif other.expression==1:
            return self
        else:
            return Expression(self.expression*other.expression)

class ComplexExpression:
    def __init__(self, real, imag):
        self.real = real
        self.imag = imag

    def multiply(self, other):
        return ComplexExpression(
            self.real.mult(other.real).subtract(self.imag.mult(other.imag)),
            self.real.mult(other.imag).add(self.imag.mult(other.real))
        )

    def add(self, other):
        return ComplexExpression(
            self.real.add(other.real),
            self.imag.add(other.imag)
        )

    def print_me(self):
        print('Real part: {}'.format(self.real))
        print('Imag part: {}'.format(self.imag))

def generate_symmetric_polynomials(list_of_complex_expression):
    z = list_of_complex_expression
    n = len(z)
    retval = []
```

```
    one = ComplexExpression(Expression(1),Expression(0))
    retval.append(one)
    truncated_sym_pols = [[one for k in range(n+1)] for ell in range(n+1)]
    for ell in range(1,n):
        truncated_sym_pols[ell][ell] = truncated_sym_pols[ell-1][ell-1].
 ↪multiply(z[ell-1])
        for k in range(ell+1,n+1):
            truncated_sym_pols[ell][k] = truncated_sym_pols[ell][k-1].
 ↪add(truncated_sym_pols[ell-1][k-1].multiply(z[k-1]))
        retval.append(truncated_sym_pols[ell][n])
    retval.append(one)

    for index in range(n):
        retval[n] = retval[n].multiply(z[index])

    return retval

Z = [
    ComplexExpression(Expression(A[index]),Expression(B[index]))
    for index in range(n)
]
W = [ComplexExpression(Expression(C[index]),Expression(D[index]))
     for index in range(n-1)
]
sym_pols_roots = generate_symmetric_polynomials(Z)
sym_pols_critical_points = generate_symmetric_polynomials(W)

real_sym_pol_constraints = [
    (n-index)*sym_pols_roots[index].real.expression
    -n*sym_pols_critical_points[index].real.expression
    for index in range(1,n)
]
imag_sym_pol_constraints = [
    (n-index)*sym_pols_roots[index].imag.expression
    -n*sym_pols_critical_points[index].imag.expression
    for index in range(1,n)
]
#this space intentionally left
```

[5]:
```
ideal_generators =\
unit_circle_constraints+\
distance_constraints+\
real_sym_pol_constraints+\
imag_sym_pol_constraints
I = Ideal(ideal_generators)
#I.groebner_basis()[8]
```

[6]:
```
#Define some helper functions
def multi_index_to_monomial(multiIndex,variableList):
    retval =1
    for i in range(0,len(multiIndex)):
        retval = retval*variableList[i]^multiIndex[i]
    return retval

import numpy as np

import collections
def order_and_print_dictionary(d):
    ordered_dictionary = collections.OrderedDict(sorted(d.items()))
    #for key in ordered_dictionary:
        #print("Key: {}  Value: {}".format(key,ordered_dictionary[key]))
```

[7]:
```
#Here we parameterize sums of squares using the parameters in Param_SOS
def get_monomials_for_sos():
    half_degree = maximal_degree_of_monomials_to_collect_sos/2
    degs_for_sos = []
    for d in range(half_degree+1):
        degs_for_sos+=list(iterator_fast(d,[1]*len(Send)))
    monomials_for_sos=[multi_index_to_monomial(d,Send) for d in degs_for_sos]
    return monomials_for_sos

monomials_for_sos = get_monomials_for_sos()

print('Monomials for sos: {}'.format(monomials_for_sos))
#print('Monomials for sos length: {}'.format(len(monomials_for_sos)))
#print('Calculated length: {} '.format(num_monomials_for_sos))

#take the parameters in Param_SOS
#and multiply them by the monomial list
#then square them.
#do this pythagoras_number_for_pim times

sos_index = 0
g_psos = 0
dictionary_of_parameters_for_sos_parameterization={}
for index_to_each_sos_poly in range(pythagoras_bound):
#design question: do I want to store any of this in a dictionary?
#as it stands, we are relying on a very large g_sos and g
    prod_poly = 0
    for mono1 in monomials_for_sos:
        dictionary_of_parameters_for_sos_parameterization[Param_SOS[sos_index]]=\
                    [mono1,index_to_each_sos_poly]
        prod_poly += Param_SOS[sos_index]*mono1
        sos_index += 1
```

```
    g_psos += prod_poly**2


#hrm...on second thought...
#it might be easier to compute the derivatives here,
#especially if we are going to solve the sdp
#using gradient or second-order techniques
# we will be solving a system of equations with:
# sums of degree two g_i's on the left (representing PSOS)
# and linear combinations of h_k's on the right (representing PIM)

order_and_print_dictionary(dictionary_of_parameters_for_sos_parameterization)
```

```
Monomials for sos: [1, a1, a2, b1, b2, c1, d1, s1, s2, t1]
```

[8]:
```
from sage.combinat.integer_vector_weighted import iterator_fast
dictionary_of_parameters_for_ideal_membership={}
def generate_parameterized_polynomial():
    index = 0
    g=0
    for poly in I.gens():
        #print("Considering polynomial ")
        #print(poly)
        #less wiggle room in degree 3?
        wiggle_room = maximal_degree_of_monomials_to_collect_ideal_membership-␣
 ↪poly.degree()
        #g+=Param[index]*poly
        #index+=1
        for wiggle_degree in range(wiggle_room):
            #print("\t Wiggle degree: {}".format(wiggle_degree))
            degs=list(iterator_fast(wiggle_degree,[1]*len(RootsAndCP)))
            #print("Degrees: ")
            #print(degs)
            monomials=[multi_index_to_monomial(d,RootsAndCP) for d in degs]
            #print("Monomials: ")
            #print(monomials)
            for monomial in monomials:
                dictionary_of_parameters_for_ideal_membership[Param[index]]=␣
 ↪[monomial,poly]
                g+=Param[index]*monomial*poly
                #print("updated g")
                #print(g)
                index+=1
                #print("\t\t Index to parameters: {}".format(index))
    return g
g_pim=generate_parameterized_polynomial()
```

8

[9]:
```
#Here we consider the 'intersection' of the two parameterizations
g_diff = g_psos-g_pim
order_and_print_dictionary(dictionary_of_parameters_for_ideal_membership)
```

[10]:
```
# process g_diff by considering the parameters of each monomial
def process_all_monomials_in_g_diff(poly,variable_set):
    #print('Processing all monomials in g_diff')
    #Leaky abstraction aka, assumption:
    #poly will have g_i*g_j*monomial_ij and also h_k * monomial_k
    max_deg_to_consider_here = poly.degree()-2
    retval1_expressions_to_zero_dict={}
    retval2_list_of_expressions_equal_to_zero=[]
    for wiggle_degree in range(max_deg_to_consider_here+1):

            #print("\t Wiggle degree: {}".format(wiggle_degree))
            degs=list(iterator_fast(wiggle_degree,[1]*len(variable_set)))
            #print("Degrees: ")
            #print(degs)
            monomials=[multi_index_to_monomial(d,variable_set) for d in degs]
            #print("Monomials: ")
            #print(monomials)
            for monomial in monomials:
                retval1 = factor_out_parameters_sos_and_im(poly,monomial)
                retval1_expressions_to_zero_dict[monomial]=retval1
                retval2_list_of_expressions_equal_to_zero.append(retval1)
    return retval1_expressions_to_zero_dict,␣
 ↪retval2_list_of_expressions_equal_to_zero

def factor_out_parameters_sos_and_im(poly,monomial):
    #print('\t\t extracting only the parameters of monomial {}'.format(monomial))
    retval1 = 0
    retval2 = np.zeros(len(R.gens()));
    expr=poly.coefficient(monomial)
    for coeff,monom in expr:
        if(monom.degree() == 1):
            #this is an h
            retval1 += coeff*monom
        if(monom.degree() == 2):
            #this is two g's or an h and some other monomial
            if monom.variables()[0] in Param_SOS:
                #two g's
                retval1 += coeff*monom
    #print('\t\t\t Constraint to be zeroed out: {}'.format(retval1))
    return retval1

ret1_dict, ret2 = process_all_monomials_in_g_diff(g_diff,Send)
```

```
order_and_print_dictionary(ret1_dict)
```

[11]:
```
#Now we use m. We want all monomials besides t_1^(2m) to have zero coefficients.
#So set all the expressions above to zero besildes one of them.
def convert_dictionary_into_ideal(dictionary):
    retval = []
    for key, value in dictionary.items():
        if T[0]^(2*m) == key:
            retval.append(value+1)
        else:
            retval.append(value)
    return retval

ideal_J_generators = convert_dictionary_into_ideal(ret1_dict)
J = R.ideal(ideal_J_generators)
#print(ideal_J_generators)
```

[12]:
```
#This paragraph is for verification in degree 2
#Recall in the document sent to Mihai Putinar, Martin Harrison, Pable Parrilo,␣
 ↪Cynthia Vinzant, et al., we find:
#\[2f = f_1+f_2-2f_5+(f_3-2a_2)f_3+(f_4-2b_2)f_4\]
#with
#\[f=c_1^2+d_1^2+\frac{s_1^2+s_2^2}{2}+t_1^2\]
#and
#\begin{align*}
#f_1&=a_1^2+b_1^2+s_1^2-1\\
#f_2&=a_2^2+b_2^2+s_2^2-1\\
#f_3&=a_1+a_2-2c_1\\
#f_4&=b_1+b_2-2d_1\\
#f_5&=(a_1-c_1)^2+(b_1-d_1)^2-1-t_1^2\\
#\end{align*}

#Tracking this down in our degree 2 polynomial.
substitute_g_pim = g_pim.subs(
#In other words, for the dictionary
#Key: h77  Value: [d1^2, b1 + b2 - 2*d1]
#h77 through h50 are parameters for monomials we're multiplying with f_4
#the non-zero monomials are going to be (b_1-b_2-2*d1)
h77=0,
#Key: h76  Value: [c1*d1, b1 + b2 - 2*d1]
h76=0,
#Key: h75  Value: [c1^2, b1 + b2 - 2*d1]
h75=0,
#Key: h74  Value: [b2*d1, b1 + b2 - 2*d1]
h74=0,
#Key: h73  Value: [b2*c1, b1 + b2 - 2*d1]
h73=0,
```

10

```
#Key: h72  Value: [b2^2, b1 + b2 - 2*d1]
h72=0,
#Key: h71  Value: [b1*d1, b1 + b2 - 2*d1]
h71=0,
#Key: h70  Value: [b1*c1, b1 + b2 - 2*d1]
h70=0,
#Key: h69  Value: [b1*b2, b1 + b2 - 2*d1]
h69=0,
#Key: h68  Value: [b1^2, b1 + b2 - 2*d1]
h68=0,
#Key: h67  Value: [a2*d1, b1 + b2 - 2*d1]
h67=0,
#Key: h66  Value: [a2*c1, b1 + b2 - 2*d1]
h66=0,
#Key: h65  Value: [a2*b2, b1 + b2 - 2*d1]
h65=0,
#Key: h64  Value: [a2*b1, b1 + b2 - 2*d1]
h64=0,
#Key: h63  Value: [a2^2, b1 + b2 - 2*d1]
h63=0,
#Key: h62  Value: [a1*d1, b1 + b2 - 2*d1]
h62=0,
#Key: h61  Value: [a1*c1, b1 + b2 - 2*d1]
h61=0,
#Key: h60  Value: [a1*b2, b1 + b2 - 2*d1]
h60=0,
#Key: h59  Value: [a1*b1, b1 + b2 - 2*d1]
h59=0,
#Key: h58  Value: [a1*a2, b1 + b2 - 2*d1]
h58=0,
#Key: h57  Value: [a1^2, b1 + b2 - 2*d1]
h57=0,
#Key: h56  Value: [d1, b1 + b2 - 2*d1]
h56=2,
#Key: h55  Value: [c1, b1 + b2 - 2*d1]
h55=0,
#Key: h54  Value: [b2, b1 + b2 - 2*d1]
h54=1,
#Key: h53  Value: [b1, b1 + b2 - 2*d1]
h53=-1,
#Key: h52  Value: [a2, b1 + b2 - 2*d1]
h52=0,
#Key: h51  Value: [a1, b1 + b2 - 2*d1]
h51=0,
#Key: h50  Value: [1, b1 + b2 - 2*d1]
h50=0,
#h49 through h22 correspond to monomials we multiply by f_3
```

```
#the non-zero monomials will be a1-a2-2*c1
#Key: h49  Value: [d1^2, a1 + a2 - 2*c1]
h49=0,
#Key: h48  Value: [c1*d1, a1 + a2 - 2*c1]
h48=0,
#Key: h47  Value: [c1^2, a1 + a2 - 2*c1]
h47=0,
#Key: h46  Value: [b2*d1, a1 + a2 - 2*c1]
h46=0,
#Key: h45  Value: [b2*c1, a1 + a2 - 2*c1]
h45=0,
#Key: h44  Value: [b2^2, a1 + a2 - 2*c1]
h44=0,
#Key: h43  Value: [b1*d1, a1 + a2 - 2*c1]
h43=0,
#Key: h42  Value: [b1*c1, a1 + a2 - 2*c1]
h42=0,
#Key: h41  Value: [b1*b2, a1 + a2 - 2*c1]
h41=0,
#Key: h40  Value: [b1^2, a1 + a2 - 2*c1]
h40=0,
#Key: h39  Value: [a2*d1, a1 + a2 - 2*c1]
h39=0,
#Key: h38  Value: [a2*c1, a1 + a2 - 2*c1]
h38=0,
#Key: h37  Value: [a2*b2, a1 + a2 - 2*c1]
h37=0,
#Key: h36  Value: [a2*b1, a1 + a2 - 2*c1]
h36=0,
#Key: h35  Value: [a2^2, a1 + a2 - 2*c1]
h35=0,
#Key: h34  Value: [a1*d1, a1 + a2 - 2*c1]
h34=0,
#Key: h33  Value: [a1*c1, a1 + a2 - 2*c1]
h33=0,
#Key: h32  Value: [a1*b2, a1 + a2 - 2*c1]
h32=0,
#Key: h31  Value: [a1*b1, a1 + a2 - 2*c1]
h31=0,
#Key: h30  Value: [a1*a2, a1 + a2 - 2*c1]
h30=0,
#Key: h29  Value: [a1^2, a1 + a2 - 2*c1]
h29=0,
#Key: h28  Value: [d1, a1 + a2 - 2*c1]
h28=0,
#Key: h27  Value: [c1, a1 + a2 - 2*c1]
h27=2,
```

```
#Key: h26  Value: [b2, a1 + a2 - 2*c1]
h26=0,
#Key: h25  Value: [b1, a1 + a2 - 2*c1]
h25=0,
#Key: h24  Value: [a2, a1 + a2 - 2*c1]
h24=1,
#Key: h23  Value: [a1, a1 + a2 - 2*c1]
h23=-1,
#Key: h22  Value: [1, a1 + a2 - 2*c1]
h22=0,
#h21 through h15 correspond to f_5
#Key: h21  Value: [d1, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h21=0,
#Key: h20  Value: [c1, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h20=0,
#Key: h19  Value: [b2, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h19=0,
#Key: h18  Value: [b1, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h18=0,
#Key: h17  Value: [a2, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h17=0,
#Key: h16  Value: [a1, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h16=0,
#Key: h15  Value: [1, a1^2 - 2*a1*c1 + b1^2 - 2*b1*d1 + c1^2 + d1^2 - t1^2 - 1]
h15=2,
#Key: h14  Value: [d1, a2^2 + b2^2 + s2^2 - 1]
h14=0,
#Key: h13  Value: [c1, a2^2 + b2^2 + s2^2 - 1]
h13=0,
#Key: h12  Value: [b2, a2^2 + b2^2 + s2^2 - 1]
h12=0,
#Key: h11  Value: [b1, a2^2 + b2^2 + s2^2 - 1]
h11=0,
#Key: h10  Value: [a2, a2^2 + b2^2 + s2^2 - 1]
h10=0,
#Key: h9  Value: [a1, a2^2 + b2^2 + s2^2 - 1]
h9=0,
#Key: h8  Value: [1, a2^2 + b2^2 + s2^2 - 1]
h8=-1,
#Key: h7  Value: [d1, a1^2 + b1^2 + s1^2 - 1]
h7=0,
#Key: h6  Value: [c1, a1^2 + b1^2 + s1^2 - 1]
h6=0,
#Key: h5  Value: [b2, a1^2 + b1^2 + s1^2 - 1]
h5=0,
#Key: h4  Value: [b1, a1^2 + b1^2 + s1^2 - 1]
h4=0,
```

```
#Key: h3  Value: [a2, a1^2 + b1^2 + s1^2 - 1]
h3=0,
#Key: h2  Value: [a1, a1^2 + b1^2 + s1^2 - 1]
h2=0,
#Key: h1  Value: [1, a1^2 + b1^2 + s1^2 - 1]
h1=-1
)
#dictionary_for_integer_solutions = get_a_solution()
#def get_degree_2_solution():
print("Substitute| values for parameters for ideal membership")
print(substitute_g_pim)
#J.groebner_basis
#print(g_pim.subs(substitutions_for_known_degree_2_certificate).subs([h4=0]))

#Here, we have a choice of how to label the squares...
substitute_g_psos = g_psos.subs(
    #let's let index 5 correspond to c1^2
#Key: g60  Value: [t1, 5]
    g60=0,
#Key: g59  Value: [s2, 5]
    g59=0,
#Key: g58  Value: [s1, 5]
    g58=0,
#Key: g57  Value: [d1, 5]
    g57=0,
#Key: g56  Value: [c1, 5]
    g56=sqrt(2),
#Key: g55  Value: [b2, 5]
    g55=0,
#Key: g54  Value: [b1, 5]
    g54=0,
#Key: g53  Value: [a2, 5]
    g53=0,
#Key: g52  Value: [a1, 5]
    g52=0,
#Key: g51  Value: [1, 5]
    g51=0,
    #ok, now let index 4 correspond to d1^2
#Key: g50  Value: [t1, 4]
    g50=0,
#Key: g49  Value: [s2, 4]
    g49=0,
#Key: g48  Value: [s1, 4]
    g48=0,
#Key: g47  Value: [d1, 4]
    g47=sqrt(2),
#Key: g46  Value: [c1, 4]
```

14

```
    g46=0,
#Key: g45  Value: [b2, 4]
    g45=0,
#Key: g44  Value: [b1, 4]
    g44=0,
#Key: g43  Value: [a2, 4]
    g43=0,
#Key: g42  Value: [a1, 4]
    g42=0,
#Key: g41  Value: [1, 4]
    g41=0,
    #ok, now let index 3 correspond to s1
#Key: g40  Value: [t1, 3]
    g40=0,
#Key: g39  Value: [s2, 3]
    g39=0,
#Key: g38  Value: [s1, 3]
    g38=1,
#Key: g37  Value: [d1, 3]
    g37=0,
#Key: g36  Value: [c1, 3]
    g36=0,
#Key: g35  Value: [b2, 3]
    g35=0,
#Key: g34  Value: [b1, 3]
    g34=0,
#Key: g33  Value: [a2, 3]
    g33=0,
#Key: g32  Value: [a1, 3]
    g32=0,
#Key: g31  Value: [1, 3]
    g31=0,
    #ok, now let index 2 correspond to s2
#Key: g30  Value: [t1, 2]
    g30=0,
#Key: g29  Value: [s2, 2]
    g29=1,
#Key: g28  Value: [s1, 2]
    g28=0,
#Key: g27  Value: [d1, 2]
    g27=0,
#Key: g26  Value: [c1, 2]
    g26=0,
#Key: g25  Value: [b2, 2]
    g25=0,
#Key: g24  Value: [b1, 2]
    g24=0,
```

```
#Key: g23  Value: [a2, 2]
    g23=0,
#Key: g22  Value: [a1, 2]
    g22=0,
#Key: g21  Value: [1, 2]
    g21=0,
    #well, I guess we have a few uncessary squares...
    #(it's 5 for the positivity certificate, 4 for the real radical certificate,
    #because one of the squares in the positivity certificate is t_1.
#Key: g20  Value: [t1, 1]
    g20=0,
#Key: g19  Value: [s2, 1]
    g19=0,
#Key: g18  Value: [s1, 1]
    g18=0,
#Key: g17  Value: [d1, 1]
    g17=0,
#Key: g16  Value: [c1, 1]
    g16=0,
#Key: g15  Value: [b2, 1]
    g15=0,
#Key: g14  Value: [b1, 1]
    g14=0,
#Key: g13  Value: [a2, 1]
    g13=0,
#Key: g12  Value: [a1, 1]
    g12=0,
#Key: g11  Value: [1, 1]
    g11=0,
#Key: g10  Value: [t1, 0]
    g10=0,
#Key: g9  Value: [s2, 0]
    g9=0,
#Key: g8  Value: [s1, 0]
    g8=0,
#Key: g7  Value: [d1, 0]
    g7=0,
#Key: g6  Value: [c1, 0]
    g6=0,
#Key: g5  Value: [b2, 0]
    g5=0,
#Key: g4  Value: [b1, 0]
    g4=0,
#Key: g3  Value: [a2, 0]
    g3=0,
#Key: g2  Value: [a1, 0]
    g2=0,
```

```
#Key: g1  Value: [1, 0]
    g1=0
)
print('Substituted values for parameters for sums of squares')
print(substitute_g_psos)
print("Sum ")
print(substitute_g_psos+substitute_g_pim)
print('Therefore, the vector of substituted values for the parameters')
print(' is a certificate of real radical membership for t1')
```

```
Substitute| values for parameters for ideal membership
-2*c1^2 - 2*d1^2 - s1^2 - s2^2 - 2*t1^2
Substituted values for parameters for sums of squares
2*c1^2 + 2*d1^2 + s1^2 + s2^2
Sum
-2*t1^2
Therefore, the vector of substituted values for the parameters
 is a certificate of real radical membership for t1
```

[ ]:

```

```

A.2 Example of YALMIP Results

A.2.0.1 Degree 2 relaxation order 2. YALMIP form

Let

$$g_0 = 1 - (x_1 - x_3)^2 - (x_2 - x_4)^2$$

$$\begin{aligned}
\gamma_1 &= 1 - x_1^2 - x_2^2 \geq 0 \\
\gamma_2 &= x_1 + 1 - 2 * x_3 \geq 0 \\
\gamma_3 &= -g_2 \geq 0 \\
\gamma_4 &= x_2 - 2 * x_4 \geq 0 \\
\gamma_5 &= -g_4 \geq 0
\end{aligned}$$

Let $\tau_i = m_1 T_i m_1^\top$ for $m_1 = [1, x_4, x_3, x_2, x_1]$

where

$$T_1 = \begin{bmatrix} 0.4932 & 0 & -0.68799 & 0 & 0.39067 \\ 0 & 2.3114 & 0 & -0.67353 & 0 \\ -0.68799 & 0 & 1.9522 & 0 & -0.56655 \\ 0 & -0.67353 & 0 & 1.0314 & 0 \\ 0.39067 & 0 & -0.56655 & 0 & 0.78813 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1.9249 & 0 & 0.62027 & 0 & -0.34678 \\ 0 & 1.6828 & 0 & 0.015398 & 0 \\ 0.62027 & 0 & 2.0654 & 0 & -0.13027 \\ 0 & 0.015398 & 0 & 1.7429 & 0 \\ -0.34678 & 0 & -0.13027 & 0 & 1.9321 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 2.3412 & 0 & -0.69918 & 0 & 0.37207 \\ 0 & 2.1517 & 0 & -0.018831 & 0 \\ -0.69918 & 0 & 2.0654 & 0 & -0.0039098 \\ 0 & -0.018831 & 0 & 2.069 & 0 \\ 0.37207 & 0 & -0.0039098 & 0 & 1.9535 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 2.1325 & 0.68367 & 0.032588 & -0.38529 & -0.012447 \\ 0.68367 & 2.0816 & 0.11724 & -0.077989 & -0.025631 \\ 0.032588 & 0.11724 & 1.9004 & -0.065515 & -0.0021826 \\ -0.38529 & -0.077989 & -0.065515 & 1.9523 & 0.065935 \\ -0.012447 & -0.025631 & -0.0021826 & 0.065935 & 1.895 \end{bmatrix}$$

$$T_5 = \begin{bmatrix} 2.1325 & -0.68368 & 0.032588 & 0.38529 & -0.012447 \\ -0.68368 & 2.0816 & -0.11724 & -0.07799 & 0.02563 \\ 0.032588 & -0.11724 & 1.9004 & 0.06551 & -0.0021836 \\ 0.38529 & -0.07799 & 0.06551 & 1.9523 & -0.065934 \\ -0.012447 & 0.02563 & -0.0021836 & -0.065934 & 1.895 \end{bmatrix}.$$

Let $\sigma_* = m_2 S_* m_2^\top$ with $m_2 = [1, x_4, x_3, x_2, x_1, x_2x_4, x_2x_3, x_2x_2, x_1x_4, x_1x_3, x_1x_2, x_1x_1]$

where

$$S_* = \begin{bmatrix} 0.92308 & 0 & -1.0477 & 0 & 0.53632 & -0.21269 & 0 & -0.37996 & 0 & -0.10088 & 0 & -0.38676 \\ 0 & 2.627 & 0 & -1.0565 & 0 & 0 & -0.290290 & 0.13196 & 0 & 0.17419 & 0 & \\ -1.0477 & 0 & 2.3256 & 0 & -0.96336 & -0.13778 & 0 & -0.11213 & 0 & -0.25273 & 0 & 0.084382 \\ 0 & -1.0565 & 0 & 1.0889 & 0 & 0 & -0.77088 & 0 & 0.099219 & 0 & 0.50785 & 0 \\ 0.53632 & 0 & -0.96336 & 0 & 0.93769 & 0.0073587 & 0 & -0.086028 & 0 & -0.66742 & 0 & 0.40138 \\ -0.21269 & 0 & -0.13778 & 0 & 0.0073587 & 2.3114 & 0 & -0.67353 & 0 & 0.16462 & 0 & 0.22004 \\ 0 & -0.29029 & 0 & -0.77088 & 0 & 0 & 1.9522 & 0 & -0.16462 & 0 & -0.80608 & 0 \\ -0.37996 & 0 & -0.11213 & 0 & -0.086028 & -0.67353 & 0 & 1.0314 & 0 & 0.23953 & 0 & 0.29393 \\ 0 & 0.13196 & 0 & 0.099219 & 0 & 3.235e-015 & -0.16462 & 0 & 2.3114 & 0 & -0.89358 & 0 \\ -0.10088 & 0 & -0.25273 & 0 & -0.66742 & 0.16462 & 0 & 0.23953 & 0 & 1.9522 & 0 & -0.56655 \\ 0 & 0.17419 & 0 & 0.50785 & 0 & 0 & -0.80608 & 0 & -0.89358 & 0 & 1.2317 & 0 \\ -0.38676 & 0 & 0.084382 & 0 & 0.40138 & 0.22004 & 0 & 0.29393 & 0 & -0.56655 & 0 & 0.78813 \end{bmatrix}$$

Then $\sigma_* = g_0 - \sum_{i=1}^{5} \tau_i \gamma_i$. All matrices above are positive semidefinite.

A.2.0.2 Degree 2 Post-processed

Let

$$g_0 = 1 - (x_1 - x_3)^2 - (x_2 - x_4)^2$$

$$\begin{aligned} g_1 &= 1 - x_1^2 - x_2^2 \geq 0 \\ g_2 &= x_1 + 1 - 2 * x_3 = 0 \\ g_3 &= x_2 - 2 * x_4 = 0 \end{aligned}$$

Let $\sigma_i = m_1 S_i m_1^\top$ for $m_1 = [1, x_4, x_3, x_2, x_1]$

Then $S_1 = T_1$, $S_2 = T_2 - T_3$, $S_3 = T_4 - T_5$. So that

$$\sigma_* = g_0 - \sum_{i=1}^{3} \sigma_i g_i$$

.

With:

$$S_2 = \begin{bmatrix} -0.4163 & 0 & 1.3195 & 0 & -0.7188 \\ 0 & -0.4689 & 0 & 0.0342 & 0 \\ 1.3195 & 0 & 0 & 0 & -0.1264 \\ 0 & 0.0342 & 0 & -0.3261 & 0 \\ -0.7188 & 0 & -0.1264 & 0 & -0.0214 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 0 & 1.3674 & 0 & -0.7706 & 0 \\ 1.3674 & 0 & 0.2345 & 0 & -0.0513 \\ 0 & 0.2345 & 0 & -0.1310 & 0 \\ -0.7706 & 0 & -0.1310 & 0 & 0.1319 \\ 0 & -0.0513 & 0 & 0.1319 & 0 \end{bmatrix}$$

A.3 Example of Gloptipoly Results

```
EDU>> cd SeDuMi_1_3\SeDuMi_1_3\; path(path,pwd); cd ..\..\gloptipoly3;

EDU>> path(path,pwd); cd ..\SOSTOOLS\SOSTOOLS.300; addsostools;

EDU>> cd ..\..\YALMIP\yalmip; addpath(genpath(pwd)); cd ..\..;

EDU>> cd Sendov\gloptipoly\

EDU>> Deg2

g0 = 1-(x(1)-x(3))^2-(x(2)-x(4))^2


Scalar polynomial


1-x(1)^2+2x(1)x(3)-x(2)^2+2x(2)x(4)-x(3)^2-x(4)^2



K= [x(1)^2+x(2)^2<=1,...

x(1)+1-2*x(3)==0,...

x(2)-2*x(4)==0]
```

```
1-by-3 vector measure support inequality
(1,1):x(1)^2+x(2)^2 <= 1
(1,2):1+x(1)-2x(3) == 0
(1,3):x(2)-2x(4) == 0

P = msdp(min(g0),K)
GloptiPoly 3.7 of 31 October 2013
Define moment SDP problem
  Valid objective function
  Number of support constraints = 3
  Number of moment constraints = 0
Measure 1
  Degree = 2
  Variables = 4
  Moments = 15
Relaxation order = 1
Mass of measure 1 set to one
Total number of moments = 15
Perform moment substitutions
Number of moments after substitutions = 14
Generate moment and support constraints
Generate moment SDP problem

Moment SDP problem
  Measure label            = 1
  Relaxation order         = 1
  Decision variables       = 14
  Linear equalities        = 10
  Linear inequalities      = 1
  Semidefinite inequalities = 5x5


[status,obj] = msol(P)
```

```
GloptiPoly 3.7 of 31 October 2013
Solve moment SDP problem
*****************************************************
Calling SeDuMi
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, Adaptive Step-Differentiation, theta = 0.250, beta = 0.500
Put 10 free variables in a quadratic cone
eqs m = 14, order n = 9, dim = 38, blocks = 3
nnz(A) = 40 + 0, nnz(ADA) = 196, nnz(L) = 105
 it :     b*y       gap    delta  rate   t/tP*  t/tD*   feas cg cg  prec
  0 :            5.67E+000 0.000
  1 :  6.63E-001 1.41E+000 0.000 0.2481 0.9000 0.9000   1.53  1  1  1.5E+000
  2 :  7.04E-001 4.25E-001 0.000 0.3024 0.9000 0.9000   1.63  1  1  4.4E-001
  3 :  9.53E-001 8.97E-002 0.000 0.2108 0.9000 0.9000   1.12  1  1  8.8E-002
  4 :  9.98E-001 3.14E-003 0.000 0.0350 0.9900 0.9900   1.03  1  1  3.1E-003
  5 :  1.00E+000 9.83E-006 0.000 0.0031 0.9990 0.9990   1.00  1  1  9.6E-006
  6 :  1.00E+000 1.01E-006 0.000 0.1027 0.9461 0.9450   1.03  1  1  9.9E-007
  7 :  1.00E+000 4.22E-008 0.000 0.0417 0.9900 0.9903   1.01  1  1  3.5E-008
  8 :  1.00E+000 2.12E-009 0.267 0.0503 0.9900 0.9900   1.02  1  1  1.8E-009

iter seconds digits       c*x               b*y
  8      2.2   8.8  1.0000000011e+000  9.9999999946e-001
|Ax-b| =  1.4e-009, [Ay-c]_+ =  1.8E-010, |x|= 4.2e+000, |y|= 1.4e+000

Detailed timing (sec)
   Pre          IPM          Post
6.250E-001    2.156E+000    2.500E-001
Max-norms: ||b||=2, ||c|| = 1,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 1003.29.
*****************************************************
Check feasibility (eps = 1.0000e-003):
  Marginally feasible SDP: residual = -1.3325e-010
Check Euclidean norm of solution (max = 1.0000e+006):
```

```
  Norm = 1.4142e+000

Check first order moments (abs tol = 1.0000e-003):

  Solution 1

    SDP objective = 5.4266e-010

    Solution reaches same objective

    Solution is feasible

Global optimality certified numerically


status =


     1




obj =


  5.4266e-010


mu = meas


Measure 1 on 4 variables: x(1),x(2),x(3),x(4)

  with moments of degree up to 2,  supported on 1 point


mv = mvec(mu)


15-by-1 moment vector


(1,1):I[1]d[1]

(2,1):I[x(1)]d[1]

(3,1):I[x(2)]d[1]

(4,1):I[x(3)]d[1]

(5,1):I[x(4)]d[1]

(6,1):I[x(1)^2]d[1]

(7,1):I[x(1)x(2)]d[1]
```

```
(8,1):I[x(1)x(3)]d[1]
(9,1):I[x(1)x(4)]d[1]
(10,1):I[x(2)^2]d[1]
(11,1):I[x(2)x(3)]d[1]
(12,1):I[x(2)x(4)]d[1]
(13,1):I[x(3)^2]d[1]
(14,1):I[x(3)x(4)]d[1]
(15,1):I[x(4)^2]d[1]

double(mv)

ans =

    1.0000
   -1.0000
   -0.0000
    0.0000
   -0.0000
    1.0000
    0.0000
   -0.0000
    0.0000
    0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
    0.0000

double(mmat(mu))

ans =
```

```
  1.0000   -1.0000   -0.0000    0.0000   -0.0000
 -1.0000    1.0000    0.0000   -0.0000    0.0000
 -0.0000    0.0000    0.0000   -0.0000    0.0000
  0.0000   -0.0000   -0.0000    0.0000    0.0000
 -0.0000    0.0000    0.0000    0.0000    0.0000
```

A.4 2012 Dissertation Draft

In this section, we include a dissertation draft from 2012, as it contains numerous figures illustrating examples related to our work.

Remark A.2. The variable encodings in "Chapter 4" include our original, full $u\mathcal{H}d$ bootstrapping.

Abstract

Zeros and Critical Points of Univariate Complex Polynomials

Richard Spjut

Conjecture 0.0.1. *For a complex polynomial of degree two or more with all its zeros contained in the closed unit disk, it remains conjectured that within a unit distance of each zero there exists a critical point.*

From this dissertation come two potential paths for a proof and three kinds of numerical experiments. The first path for a proof we call 'one zero out and back,' or OZOAB. The second path for a proof we call 'acceptable constants of integration,' or ACOI. The first two kinds of numerical experiments are those that support the first two paths for a proof. The final kind of numerical experiments is related to searches of my predecessors for extremal polynomials.

Contents

List of Figures

Chapter 1

Introduction

The earliest date attributed to Conjecture 0.0.1 is 1958. In 1967, Hayman provided more publicity for the conjecture. Since that time, the conjecture has been proved in the following cases.

In 1968, Brannan proved the conjecture for polynomials of degree three, Phelps and Rodriguez proved the conjecture for polynomials of degree four, and Rubinstein proved that a zero of unit modulus will have a critical point within unit distance. One year later, Meir and Sharma proved the conjecture for polynomials of degree five and Schmeiser proved the conjecture for polynomials with at least one of the zeros at the origin. In 1971, Saff and Twomey proved the conjecture for polynomials with precisely three distinct zeros and Schmeisser proved the conjecture for monic polynomials with non-positive, real coefficients. One year later, Phelps and Rodriguez proved the conjecture for polynomials with only real zeros. In 1985, Bojanov, Rahman, and Syznal proved that within distance 1.08006 of each zero there exists a critical point, and that as the degree of the polynomial tends to infinity this upper bound tends to 1. In 1994 and in separate papers, Borcea and Katsoprinakis proved the conjecture for polynomials of degree six. In 1996, Borcea proved the conjecture for polynomials of degree seven. In 1999, Brown and Xiang proved the conjecture for polynomials of degree eight. In 2004, Rahman and Schmeisser proved that within distance 1.075 of each zero there is a critical point.

Corollary 0.0.1 resembles a corollary of the Gauss-Lucas theorem, namely, within a unit distance of each critical point, there is a zero.

For further history and references see [**?**].

Chapter 2

One Zero Out And Back

Notation 2.0.2. Let $f(z)$ be a univariate complex polynomial with zeros contained in the closed unit disk. Respectively, denote its zeros and critical points as $\{z_k\}_{k=1}^{n}$ and $\{w_j\}_{j=1}^{n-1}$. For one $a \in \mathbb{C}$

$$f(z) = a \prod_{k=1}^{n} (z - z_k)$$

and

$$\frac{\partial f}{\partial z} = a \sum_{j=1}^{n} \prod_{k=1 k \neq j}^{n} (z - z_k) = an \prod_{\ell=1}^{n-1} (z - w_\ell).$$

The following three lemmas are classical.

Lemma 2.0.3. *Equate the second leading coefficient, and obtain*

$$\frac{1}{n} \sum_{k=1}^{n} z_k = \frac{1}{n-1} \sum_{j=1}^{n-1} w_j.$$

That is, the centroid of the zeros and the centroid of the critical points coincide.

Lemma 2.0.4.

$$0 = \sum_{k=1}^{n} \frac{1}{w_j - z_k} \tag{2.1}$$

Proof. Consider the logarithmic derivative. See pg. 6 of [**?**] $\square$

Lemma 2.0.5. *Every critical point is a convex combination of zeros.*

Proof. If $w_j = z_k$ for some k we are done. Otherwise let

$$\lambda_k = \frac{1/|w_j - z_k|^2}{\sum_{\alpha=1}^{n} 1/|w_j - z_\alpha|^2}$$

$$w_j = \sum_{k=1}^{n} \lambda_k z_k$$

$\square$

Notation 2.0.6. When $k \geq 2$, inductively define the fundamental symmetric polynomials in $\{z_1, z_2, \ldots, z_k\}$.

$$\begin{array}{lll}
 & sym_0(\{z_1,\ldots,z_k\}) & = 1 \\
 & sym_1(\{z_1,\ldots,z_k\}) & = \sum_{j=1}^{k} z_j \\
if(\ell = k) & sym_\ell(\{z_1,\ldots,z_k\}) & = \prod_{j=1}^{k} z_\ell \\
if(\ell > k) & sym_\ell(\{z_1,\ldots,z_k\}) & = sym_{\ell-1}(\{z_1,\ldots,z_{k-1}\})z_k + sym_\ell(\{z_1,\ldots,z_{k-1}\})
\end{array}$$

So that

$$f(z) \quad = a \textstyle\sum_{\ell=0}^{n} sym_\ell(\{z_1,\ldots,z_n\}) z^{n-\ell} \tag{2.2}$$

$$\frac{\partial f}{\partial z} = a \sum_{\ell=0}^{n-1} (n-\ell) sym_\ell(\{z_1,\ldots,z_n\}) z^{n-\ell-1} \tag{2.3}$$

$$= an \sum_{j=0}^{n-1} sym_j(\{w_1,\ldots,w_{n-1}\}) z^{n-1-j} \tag{2.4}$$

Generalize Lemma 2.0.3 by equating coefficients in equations 2.3 and 2.4.

Lemma 2.0.7. *For each $1 \leq \ell \leq n-1$,*

$$\frac{1}{n} sym_\ell(\{z_1, \ldots, z_n\}) = \frac{1}{n-\ell} sym_\ell(\{w_1, \ldots, w_{n-1}\})$$

Lemma 2.0.8 (Rubinstein, 1968)**.** *If a zero is on the boundary, then it contains a critical point within unit distance.*

Proof. Without loss of generality,

1) $z_1 = 1$

2) $z_k \neq 1$ for $k = 2, 3, \ldots n$

   and

3) $f'(1) = 1$.

Towards contradiction, for $|z| < 1$, $f'(z+1) = (1-zh(z))^{n-1}$ where $h(z)$ is analytic in the open unit disc and $|h| < 1$. By differentiation, $f''(1) = (1-n)h(0)$. The polynomial $Q(z) \equiv f(z)/(z-1)$ satisfies $Q(1) = f'(1) = 1$ and $2Q'(1) = f''(1)$, so that $|Q'(1)| < \frac{n-1}{2}$. However, consider the logarithmic derivative.

$$Q'(1) = \frac{Q'(1)}{Q(1)} = \sum_{k=2}^{n} \frac{1}{1-z_k}$$

$|z_k| \leq 1 \Rightarrow Re[1/(1-z_k)] \geq \frac{1}{2}$ and thus, $Re[Q'(1)] \geq \frac{n-1}{2}$.

$\square$

2.1 Sending One Zero Out

Suppose that we fix all of the zeros besides one. WLOG let this be z_n. Let the polar decompostion be $z_n = R_n e^{i\theta_n}$, and then provide a one-parameter family: $z = Re^{i\theta_n}$ with $R \in [R_n, \infty)$. Here we are 'sending one zero out.'

Remark 2.1.1. As $R \to \infty$ or $R >> 1$, we notice that one critical point 'comes for the ride.' Suppose $|w| >> 1$ and then $(w - z_k) \sim w$ for $k \neq n$. Equation 2.1 becomes:

$$0 \sim \frac{1}{w - z_n} + \frac{n-1}{w}$$

$$w \sim (n-1)(z_n - w)$$

$$w \sim \frac{n-1}{n} z_n$$

Formally, lemmas 2.1.4 and 2.0.5 confirm this remark.

Lemma 2.1.2. *For sufficiently large R there is an M for which there exists a unique critical point outside of a circle centered at the origin of radius M.*

Proof. If $n = 2$ the result is obvious with $M = 1$ and $R > 3$. Assume $n \geq 3$.

Consider Equation 2.3. When we prove for R large enough that

$$|(n-1)sym_1(\{z_1, \ldots, z_n\})|M^{n-2} > nM^{n-1} + \sum_{\ell=2}^{n-1} |(n-\ell)sym_\ell(\{z_1, \ldots, z_n\})|M^{n-\ell-1}$$

we can conclude by Rouche's theorem that there are $n-2$ critical points in the circle $|z| = M$.

First bound the left side from below.

$$|(n-1)sym_1(\{z_1,\ldots,z_n\})| = |(n-1)(z_1+\ldots+z_n)|$$

$$\geq (n-1)(R-|z_1+\ldots+z_{n-1}|) \geq (n-1)(R-(n-1)).$$

Then bound the right side from above.

$$nM^{n-1} + \sum_{\ell=2}^{n-1} |(n-\ell)sym_\ell(\{z_1,\ldots,z_n\})|M^{n-\ell-1} =$$

$$nM^{n-1} + \sum_{\ell=2}^{n-1} |(n-\ell)(z_n sym_{\ell-1}(\{z_1,\ldots,z_{n-1}\}) + sym_\ell(\{z_1,\ldots,z_{n-1}\}))|M^{n-\ell-1} \leq$$

$$nM^{n-1} + \sum_{\ell=2}^{n-1} (n-\ell)\left(R\binom{n-1}{\ell-1} + \binom{n-1}{\ell}\right) M^{n-\ell-1}$$

Compare the two bounds and, for ease of exposition, use $M = R/2$.

$$(n-1)(R-(n-1))\left(\frac{R}{2}\right)^{n-2} >$$

$$n\left(\frac{R}{2}\right)^{n-1} + \sum_{\ell=2}^{n-1} (n-\ell)\left(R\binom{n-1}{\ell-1} + \binom{n-1}{\ell}\right)\left(\frac{R}{2}\right)^{n-\ell-1}$$

Collect terms according to powers of R,

$$(2(n-1)-n)\left(\frac{R}{2}\right)^{n-1} >$$

$$2(n-1)^2\left(\frac{R}{2}\right)^{n-2} + \sum_{\ell=2}^{n-1} (n-\ell)\left(R\binom{n-1}{\ell-1} + \binom{n-1}{\ell}\right)\left(\frac{R}{2}\right)^{n-\ell-1}$$

and it becomes clear that the all of the above inequalities hold for large R. $\square$

Lemma 2.1.3. *For sufficiently large R, there will be, including multiplcity, $n-2$ critical points contained within the circle $|z| = M = 2n$.*

Proof. Set $M = 2n$ in the proof of 2.1.2 and collect R on the left side to obtain:

$$R\left((n-1)M^{n-2} - \sum_{\ell=2}^{n-1}(n-\ell)\binom{n-1}{\ell-1}M^{n-\ell-1}\right) \geq$$
$$(n-1)^2M^{n-2} + nM^{n-1} + \sum_{\ell=2}^{n-1}(n-\ell)\binom{n-1}{\ell}M^{n-\ell-1}$$

□

Lemma 2.1.4. *For sufficiently large R there will be, including multiplicity, $n-2$ critical points contained within an epsilon distance from the unit circle.*

Proof. The bounds in the proof of Lemma 2.1.2 and 2.1.3 are poor indeed, but they allow us to make the assumption that for $n-2$ critical points, $|w| << R$, when R is large.

If $w = z_k$ for $1 \leq k \leq n-1$ then we are done. Otherwise:

$$w = \sum_{k=1}^{n} \lambda_k z_k = \frac{1/|w-z_n|^2}{\sum_{\alpha=1}^{n} 1/|w-z_\alpha|^2} z_n + \sum_{k=1}^{n-1} \lambda_k z_k$$

and we notice $\lambda_n z_n \to 0$ and $\lambda_n \to 0$ as $R \to \infty$, so that

$$w \sim \sum_{k=1}^{n-1} \lambda_k z_k$$

and

$$
\begin{aligned}
|w| &\le \lambda_n R + \sum_{k=1}^{n-1} \lambda_k = \lambda_n R + (1-\lambda_n) \\
&= 1 + \lambda_n (R-1) \le 1 + \frac{1/(R^2-1)}{\sum_{k=1}^{n-1} 1/|w-z_k|^2}(R-1) \\
&= 1 + \frac{A}{R+1}
\end{aligned}
$$

By Lemma 2.1.3, $A = \frac{1}{\sum_{k=1}^{n-1} 1/|w-z_k|^2}$ can be bounded by

$$|w - z_k| \le 2n+1 \Rightarrow \frac{1}{(2n+1)^2} \le \frac{1}{|w-z_k|^2} \Rightarrow \frac{n-1}{(2n+1)^2} \le \sum_{k=1}^{n-1} 1/|w-z_k|^2$$

$$A \le \frac{(2n+1)^2}{n-1}.$$

In other words, the critical points that get left behind eventually feel little influence by changes in z_n.

$\square$

Consider the following examples.

Example 2.1.5. Let $f(z) = (z-R)(z-1)^{n-1}$. Then

$$f'(z) = (z-1)^{n-1} + (z-R)(n-1)(z-1)^{n-2} = (z-1)^{n-2}((z-1)+(n-1)(z-R)).$$

There will be a critical point at 1 with multiplicity $n-2$ and one critical point at

$$w = \frac{(n-1)R+1}{n}$$

Example 2.1.6. Let $f(z) = (z-R)(z+1)^{n-1}$. Then

$$f'(z) = (z+1)^{n-1} + (z-R)(n-1)(z+1)^{n-2} = (z+1)^{n-2}((z+1)+(n-1)(z-R))$$

There will be a critical point at -1 with multiplicity $n-2$ and one critical point at

$$w = \frac{(n-1)R-1}{n}$$

Example 2.1.7. Let $f(z) = (z-R)(z^2+1)$. Then

$$f'(z) = (z^2+1) + (z-R)(2z) = 3z^2 - 2Rz + 1$$

$$w = \frac{R \pm \sqrt{R^2-3}}{3}$$

Example 2.1.8. Let $0 < t << 1$. I.e., let t be small and positive. Let $f(z) = (z-e^{it})(z-e^{-it})(z-R)$. Then for some values of small $R > 1$, no critical points will be within the unit circle.

Theorem 2.1.9. *Uniqueness of Critical Point that Comes For the Ride*

Proof. From Lemma 2.1.4, the fundamental theorem of algebra, and the pigeon hole principle, it follows that there is one critical point away from the unit circle.

□

Lemma 2.1.10. *Location of Critical Point*

For large R, the critical point will

(1) be 'between the centroid and $Re^{i\theta_n}$';

(2) have, within epsilon, the same angle of z_n;

(3) have a modulus within the interval $[\frac{n-1}{n}R - \frac{1}{n}, \frac{n-1}{n}R + \frac{1}{n}]$.

2.2 Critical Point Coming Along

Once we are convinced that a unique critical point 'tags along' with the zero going out, we want to determine the relative travel of that critical point with respect to changes in R.

Notice that if we let $w = re^{i\varphi}$, then implicitly differentiate Equation 2.1, we get:

$$0 = ((e^{i\varphi}dr + ire^{i\varphi}d\varphi) - (e^{i\theta_n}dR))\frac{-1}{(w-z_n)^2} + (e^{i\varphi}dr + ire^{i\varphi}d\varphi)\sum_{k=1}^{n-1}\frac{-1}{(w-z_k)^2}$$

Now, we want to isolate an expression for dr/dR.

$$\frac{e^{i\theta}dR}{(w-z_n)^2} = \sum_{k=1}^{n}\frac{1}{(w-z_k)^2}(e^{i\varphi}dr + ire^{i\varphi}d\varphi)$$

$$e^{i(\theta-\varphi)}\frac{dR}{(w-z_n)^2} = \sum_{k=1}^{n}\frac{1}{(w-z_k)^2}(dr + ird\varphi)$$

$$e^{i(\theta-\varphi)}\frac{1}{(w-z_n)^2} = \sum_{k=1}^{n}\frac{1}{(w-z_k)^2}\left(\frac{dr}{dR} + ir\frac{d\varphi}{dR}\right)$$

$$e^{i(\theta-\varphi)}\frac{1}{(w-z_n)^2\left(\sum_{k=1}^{n}\frac{1}{(w-z_k)^2}\right)} = \frac{dr}{dR} + ir\frac{d\varphi}{dR}$$

Taking the real part, it follows that:

$$\frac{dr}{dR} = Real\left[e^{i(\theta-\varphi)}\frac{1}{(w-z_n)^2\left(\sum_{k=1}^{n}\frac{1}{(w-z_k)^2}\right)}\right] \tag{2.5}$$

Accordingly,

$$\left|\frac{dr}{dR}\right| \leq \frac{1/|w-z_n|^2}{\left|\sum_{k=1}^{n}\frac{1}{(w-z_k)^2}\right|} \tag{2.6}$$

Example 2.2.1. Verify Equation 2.5 for Example 2.1.5. Given

$$w = \frac{(n-1)R+1}{n}$$

it easily follows that

$$\frac{dr}{dR} = \frac{n-1}{n}.$$

In the context of Example 2.1.5, Equation 2.5 states:

$$\begin{aligned}\frac{dr}{dR} &= \frac{1/(w-R)^2}{1/(w-R)^2 + \sum_{k=1}^{n-1} 1/(w-1)^2} \\ &= \frac{1}{1+(w-R)^2(n-1)/(w-1)^2} = \frac{1}{1+1/(n-1)} = \frac{n-1}{n}\end{aligned}$$

Example 2.2.2. Verify Equation 2.5 for Example 2.1.7. Given

$$w = \frac{R+\sqrt{R^2-3}}{3}$$

it easily follows that

$$\frac{dr}{dR} = \frac{1}{3}\left(1+\frac{R}{\sqrt{R^2-3}}\right).$$

In the context of Example 2.1.7, Equation 2.5 states:

$$\begin{aligned}\frac{dr}{dR} &= Real\left[\frac{1/(w-R)^2}{1/(w-R)^2+1/(w+i)^2+1/(w-i)^2}\right] \\ &= \frac{1}{1+2(w-R)^2(w-1)^2/(w^2+1)^2} = \frac{(w^2+1)^2}{(w^2+1)^2+2(w-R)^2(w-1)^2}\end{aligned}$$

2.3 Return

By 'sending one zero out' we have located a critical point and ascertained its dynamics. The idea now is to 'bring back' the zero and to keep track of its corresponding critical point.

Remark 2.3.1. For small epsilon, for some $R^* \in [n-1-\epsilon, n+1+\epsilon]$, the critical point will be exactly unit distance from z_n. When $0 < \frac{dr}{dR} \leq 1$, as the zero returns ($R < R^*$) the critical point becomes closer to the zero. Therefore, the critical point is within unit distance from z_n during this interval. When $|\frac{dr}{dR}| > 1$ a singularity is usually near.

As we see in numerous examples, including Example 2.1.8, for some values of R ($R \in [R_n, 1+\delta]$ where δ is) the derivative might have a double root (or multiple roots) corresponding to the critical point 'coming along.' This suggests that we might, under circumstances usually involving symmetry, lose uniqueness of the critical point when the zero returns to the unit circle.

Conjecture 2.3.2. *In vague words, the critical point that 'comes for the ride' is, 'up to bifurcation,' one of the critical points conjectured to exist within unit distance of z_n in Conjecture 0.0.1.*

Now, the issue is, how to deal with these bifurcations. There are at least two ideas.

(I) Modify the path of z_n by introducing a second parameter, θ, to avoid pairs of (R, θ) for which the corresponding critical point bifurcates. Since there are only finite number of zeros, there will be only finite combinations of (R, θ) for which bifurcation occurs, therefore we only need to 'wiggle θ' a small amount to avoid bifurcations.

(II) Allow bifurcations and define

$$h(R) = min_j |z_n - w_j|$$

Then h is continuous with respect to R. h is differentiable except at bifurcation points. It seems the L_1 norm of dh/dR can be bounded near singularities, and the L_∞ norm of h is bounded elsewhere.

2.3.1 Introducing θ

The singular behavior at bifurcations, motivates our consideration of the two-parameter family $z_n = Re^{i\theta}$. Revist our calculations to find the analog of Equation 2.5.

$$0 = \frac{1}{(w - z_n)^2}((e^{i\varphi}dr + ire^{i\varphi}d\varphi) - (e^{i\theta}dR + iRe^{i\theta}d\theta)) + \sum_{k=1}^{n-1} \frac{1}{(w - z_k)^2}(e^{i\varphi}dr + ire^{i\varphi}d\varphi)$$

we rewrite:

$$\frac{e^{i\theta}dR + iRe^{i\theta}d\theta}{(w - z_n)^2} = \sum_{k=1}^{n} \frac{1}{(w - z_k)^2}(e^{i\varphi}dr + ire^{i\varphi}d\varphi)$$

Recall, our goal is to isolate $\frac{dr}{dR}$

$$e^{i(\theta-\varphi)}\frac{dR + id\theta}{(w - z_n)^2} = \sum_{k=1}^{n} \frac{1}{(w - z_k)^2}(dr + ird\varphi)$$

$$e^{i(\theta-\varphi)}\frac{1 + i\frac{d\theta}{dR}}{(w - z_n)^2} = \sum_{k=1}^{n} \frac{1}{(w - z_k)^2}\left(\frac{dr}{dR} + ir\frac{d\varphi}{dR}\right)$$

$$e^{i(\theta-\varphi)} \frac{1 + i\frac{d\theta}{dR}}{(w-z_n)^2 \left(\sum_{k=1}^{n} \frac{1}{(w-z_k)^2} \right)} = \frac{dr}{dR} + ir\frac{d\varphi}{dR}$$

Taking the real part, it follows that:

$$\frac{dr}{dR} = Real \left[e^{i(\theta-\varphi)} \frac{1 + i\frac{d\theta}{dR}}{(w-z_n)^2 \left(\sum_{k=1}^{n} \frac{1}{(w-z_k)^2} \right)} \right] \tag{2.7}$$

2.4 No Return

We see that the behavior of the critical point upon the zero's return is difficult to assess. So, perhaps it's best to ask another question instead: At what value of R is the zero actually a unit distance away from the critical point that comes along?

Note, for large $R >> 1$, we know the critical point is of distance R/n away from the zero.

So, when $R \sim n$ we know the zero roughly escapes the critical point.

Now, if $dr/dR < 1$ the whole time, we know z_n is moving away from the critical point the whole time. Therefore, if it takes all of the path from $R_n to R = n$ to 'escape,' then it must have been closer in the first place. [idea...maybe not R wrt origin, but radius w.r.t. center]

Unfortunately this is not always the case, as in the (conjectured extremal) currently known best bound $f(z) = z^n - 1$. Here, when $R = R_n = 1$ it is the case that $dr/dR >> 1$ and the critical point races towards the zero. We want to say something about $dR > \int dr/dRdr$ during the period when $dr/dR > 1$

Perhaps what we want to say is that this only occurs when another zero is 'nearby' and the existence of a zero nearby implies the critical point is between them.

It also seems that whenever $dr/dR > 1$ and $d^2r/dR^2 > 0$ it is also the case that the 'rate of change of the distance between the nearest critical point(s) and the zero' is decreasing.

2.4.1 Multiple Humps

We can create scenarios where dr/dR becomes larger than 1 as many times as we want, that is, dr/dR will have multiple minima and maxima. We do this by creating pairs of zeros that the 'zero going out' must pass between. It seems the case that the convex hull of any three zeros contains a critical point.

2.5 Numerical Experiments

2.5.1 Introduction

In this section, critical points are found by computing the spectra of the companion matrix. This technique is prone to numerical error, but is sufficient for fast computations in which we wish to explore the dynamics of OZOAB. We step R between R_n and a fixed value, usually $R = n$. We plot dr/dR, dh/dR

Code

2.5.2 Miller's Polynomials

On page 31 of Miller thesis, counter examples to a conjecture are provided for even degrees 6 through 12. We revisit these counterexamples. Note, these are not the roots of unity, so it avoids numerical instability - which likely Miller encountered.

Degree 6

BorceaOneZeroOutAndBackTracingPaths([.84,roots([1,1.182303183,1.340070024,1.34007002

Equivalent to BorceaOneZeroOutAndBackTracingPaths([.84,-1,-.506699+.862122*i,-.506699-.862122*i,.415548+.909571*i,.415548-.909571*i],1)

Degree 8

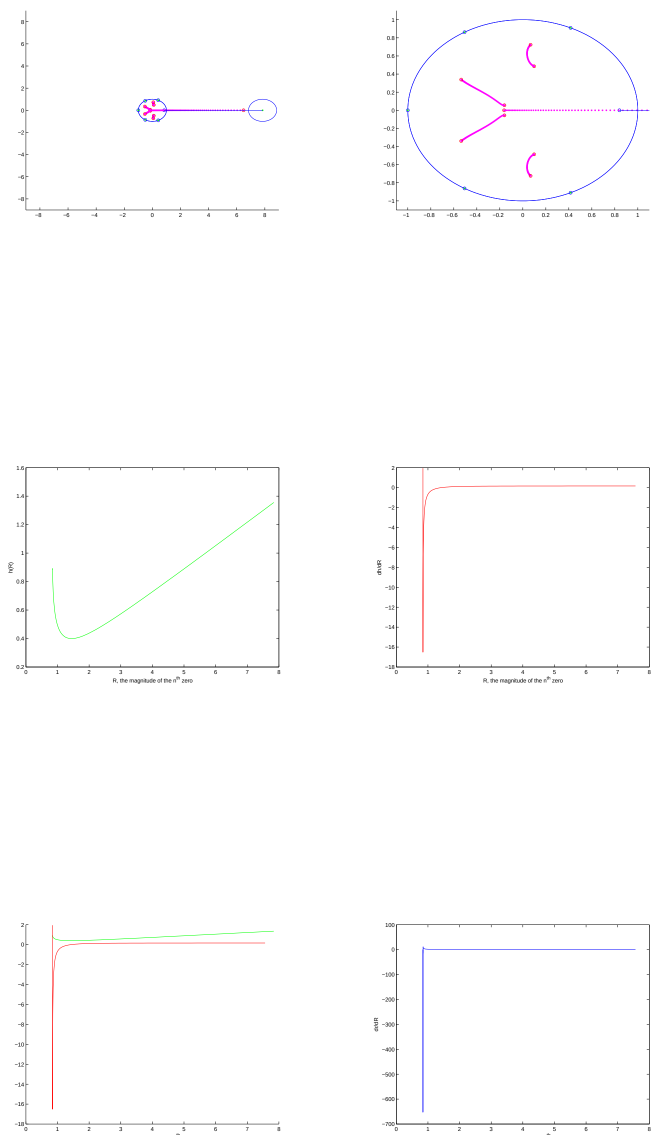
BorceaOneZeroOutAndBackTracingPaths([.8,roots([1,1.241776468,1.504033112,1.702664563,

-1.0000 -0.7166 + 0.6975i -0.7166 - 0.6975i -0.0124 + 0.9999i -0.0124 - 0.9999i 0.6081 + 0.7939i 0.6081 - 0.7939i

Degree 10

BorceaOneZeroOutAndBackTracingPaths([.7,roots([1,1.401329769,1.873192265,2.303236288,

-1.0000 -0.8159 + 0.5781i -0.8159 - 0.5781i -0.3291 + 0.9443i -0.3291 - 0.9443i 0.6775 + 0.7355i 0.6775 - 0.7355i 0.2669 + 0.9637i 0.2669 - 0.9637i



18

Figure 2.1: Miller's Degree 6 Counterexample. Clockwise from top left: The critical points

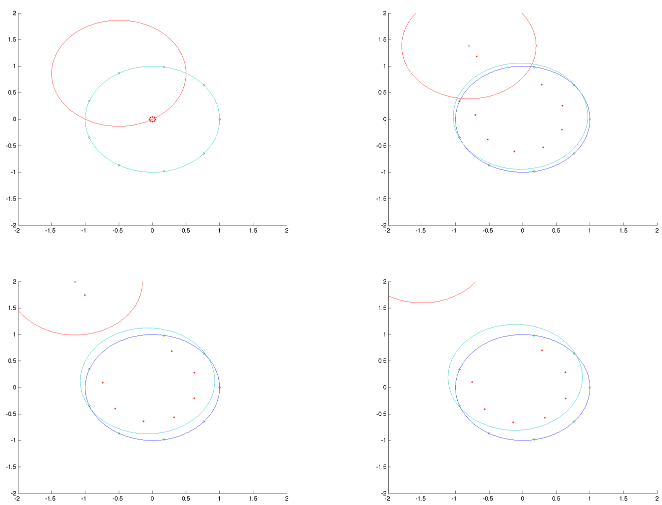


Figure 2.2: Degree9RootsOfUnity

Degree 12

BorceaOneZeroOutAndBackTracingPaths([.6,roots([1,1.589848892,2.341118841,3.1133551299

results of roots: -1.0000 -0.8714 + 0.4905i -0.8714 - 0.4905i -0.5198 + 0.8543i -0.5198 - 0.8543i -0.0338 + 0.9994i -0.0338 - 0.9994i 0.7099 + 0.7043i 0.7099 - 0.7043i 0.4202 + 0.9074i 0.4202 - 0.9074i

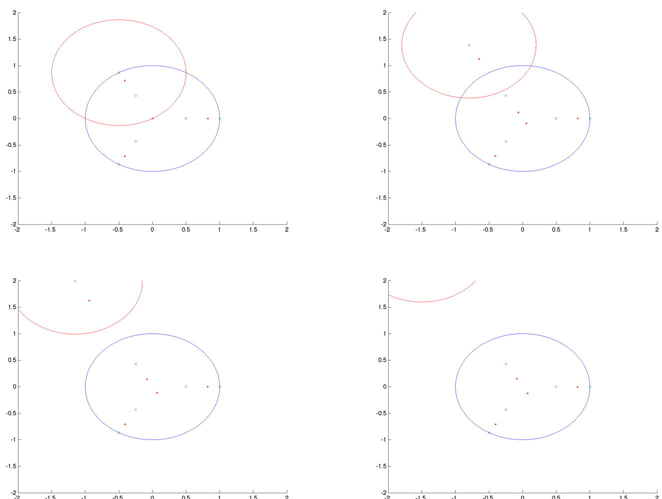


Figure 2.3: TwoSetsOf3RootsOfUnity

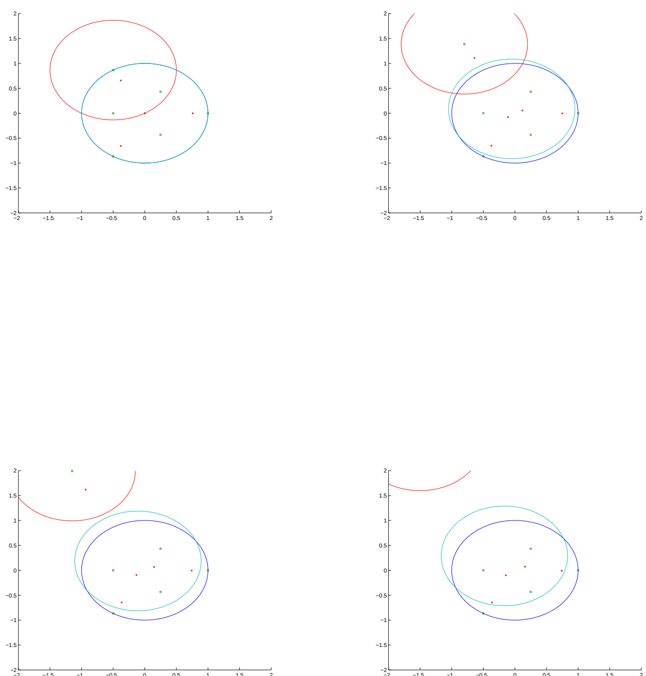


Figure 2.4: TwoSetsOf3RootsOfUnityRotated

21

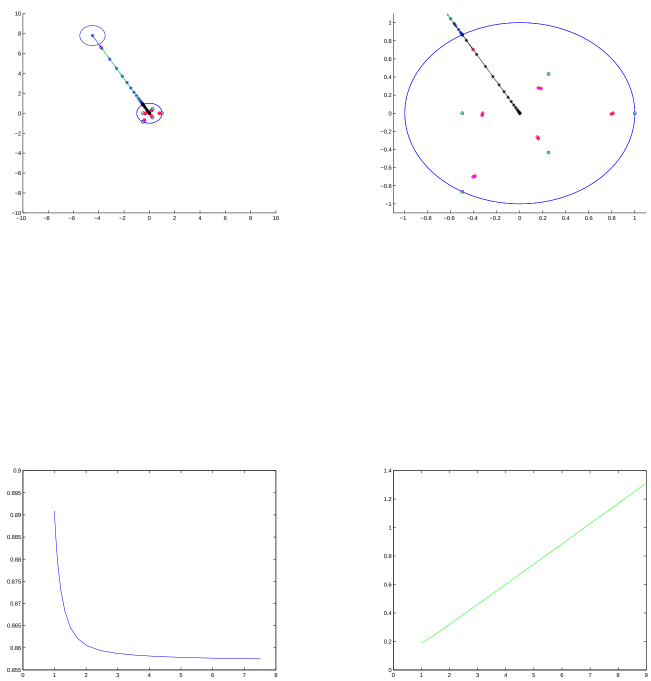


Figure 2.5: TracingPaths-TwoSetsOfRootsOfUnityAndZeroOuter

Chapter 3

Acceptable Constants Of Integration

The idea is to start with the critical points. From what we know (Brown and Xiang) we must have at least 9 distinct zeros (this does not mean we necessarily need 8 distinct critical points).

So we start with 8 critical points. Now fix the critical points. Antidifferentiate and then ask, what arbitrary constants allow for all the zeros to be contained within the unit disk?

$$f'(z) = n\prod_{j=1}^{n-1}(z - w_k) = n\sum_{\ell=0}^{n-1} sym_\ell(\{w_1, \ldots, w_{n-1}\})z^{(n-1)-\ell}$$

$$f(z) = \int f'(z)dz + C = \prod_{j=1}^{n-1} \frac{b_j}{j+1} z^{j+1}$$

We notice that $C = \prod_{k=1}^{n} z_k$.

Lemma 3.0.1. *The symmetric moments of the zeros and critical points are equal.*

Proof. Equate coefficients in the expressions:

$$f'(z) = n \sum_{\ell=0}^{n-1} sym_\ell(\{w_1, \ldots, w_{n-1}\}) z^{(n-1)-\ell} = \sum_{\ell=0}^{n-1} (n-\ell) sym_\ell(\{z_1, \ldots z_n\}) z^{n-\ell-1}$$

$$\frac{1}{n} sym_\ell(\{z_1, \ldots, z_n\}) = \frac{1}{n-\ell} sym_\ell(\{w_1, \ldots, w_{n-1}\})$$

□

Define the set of acceptable constants of integration to be

$$\mathcal{A}(\{w_j\}_{j=1}^{n-1}) = \{C | f(z) = 0 \Rightarrow |z| \leq 1\}$$

When the critical points are implied, we abreviate to $\mathcal{A}$.

Lemma 3.0.2. *$\mathcal{A}$ is contained in the unit disk.*

Proof. The bound is

$$|C| = \prod_{k=1}^{n} |z_k| \leq 1.$$

□

Furthermore, $\mathcal{A}$ is star-shaped.

Example 3.0.3. Let $w_1 = (1-i)/2$. Then $f'(z) = 2z - (1-i)$. $f(z) = z^2 - (1-i)z + C$. Notice $0 \notin \mathcal{A}(\{\frac{1-i}{2}\})$. The roots of the above quadratic are:

$$\frac{(1-i) \pm \sqrt{(1-i)^2 - 4C}}{2}$$

For which we require:

$$\left| \frac{(1-i) \pm \sqrt{-2i - 4C}}{2} \right| \leq 1$$

$$\left|(1-i) \pm \sqrt{-2i-4C}\right| \leq 2$$

$$b = -2 - 4Im[C]$$

$$a = -4Re[C]$$

$$d = \pm\sqrt{\frac{-a+\sqrt{a^2+b^2}}{2}}$$

$$d = \pm\sqrt{\frac{4Re[C]+\sqrt{(4Re[C])^2+(2+4Im[C])^2}}{2}} = \pm\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}$$

$$c = \frac{b}{2d} = \pm\frac{b}{2\sqrt{\frac{-a+\sqrt{a^2+b^2}}{2}}}$$

$$\left|(1-i) \pm \left(\frac{-1-2Im[C]}{\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}} + \sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C$$

So either

$$\left|(1-i) + \sqrt{-2i-4C}\right| \leq 2$$

$$\left|(1-i) + \left(\frac{-1-2Im[C]}{\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}} + \sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C$$

$$\left(1 - \frac{1+2Im[C]}{\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}}\right)^2 +$$

$$\left(\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}} - 1\right)^2$$

$$\leq 4$$

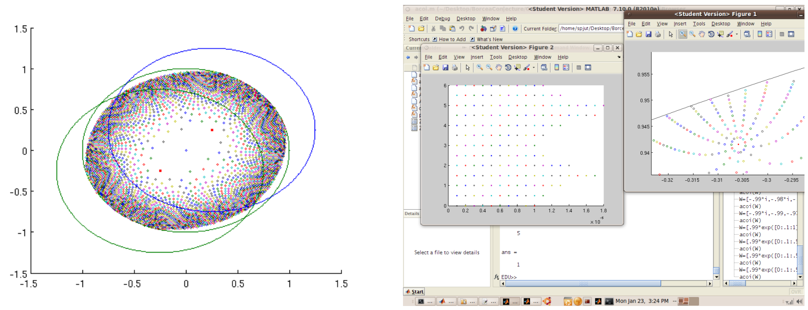


Figure 3.1: Numerical Experiments related to ACOI.

$$1-2\frac{1+2Im[C]}{\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}}+\frac{(1+2Im[C])^2}{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}+$$

$$2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}-2\sqrt{2Re[C]+\sqrt{(2Re[C])^2+(1+2Im[C])^2}}+1$$

$$\leq 4$$

or

$$\left|(1-i)-\sqrt{-2i-4C}\right|\leq 2$$

3.1 Numerical Experiments

As one can see, to compute or parametrize a region of ACOI is quite an undertaking. Software was developed to numerically compute regions of ACOI for exploratory purposes.

Chapter 4

Extremal Polynomials

In this chapter, the novel approach is to apply duality methods in optimization to Sendov's conjecture.

Over families of polynomials, we optimize the unsymmetrized Hausdorff distance [cf [**?**] §2 pg. 4, h(F,F')].

We modify the objective function to obtain results related to Borcea's conjectures.

4.1 Variables and Semi-definite Constraints

4.1.1 Real and Imaginary Parts of Zeros and Critical Points

For $1 \leq \ell \leq 6n-5$, let $x_\ell \in \mathbb{R}$.

Notation 4.1.1. For $1 \leq k \leq n$ and $1 \leq j \leq n-1$ let

$$x_{2k-1} = Real(z_k)$$

$$x_{2k} = Imag(z_k)$$

$$x_{2n+2j-1} = Real(w_j)$$

$$x_{2n+2j} = Imag(w_j)$$

so that

$$z_k = x_{2k-1} + x_{2k}i$$

$$w_j = x_{2n+2j-1} + x_{2n+2k}i$$

Semi-definite Constraints

There are n inequality constraints which ensure the zeros are contained within the closed unit disk.

$$|z_k| \leq 1$$

$$x_{2k-1}^2 + x_{2k}^2 - 1 \leq 0$$

In order to ensure that the $x_{2n+\ell}$ for $1 \leq \ell \leq 2n-2$ correspond to critical points, we use two sets of pairs of equality constraints.

One set of constraints ensures that each $x_{2n+2j-1} + x_{2n+2j}i$ is a critical point. This results in $n-1$ pairs of equality constraints, one pair for each critical point, therefore $2n-2$ equality constraints. Each pair consists of one constraint for the real part and one for the imaginary part of the equation $f'(w_j) = 0$ obtained by considering the central form of Notation 2.0.2.

The other set of constraints ensures that there is a bijection between the set $\{x_{2n+2j-1} + x_{2n+2j}i\}_{j=1}^{n-1}$ and $\{w_j\}_{j=1}^{n}$. That is, each critical point will be represented. This results in $n-1$ pairs of equality constraints, one pair for each symmetric moment besides the trivial one corresponding to nz^{n-1}, therefore $2n-2$ equality constriants. Each pair consists of one constraint for the real part and one for the imaginary part of the last equation in the proof of Lemma 3.0.1.

Note that when both sets of constraints are used, the critical points are overdetermined.

4.1.2 Unsymmetrized Hausdorff Distance

We recognize that, because there is symmetry in the zero set, if we are maximizing the Unsymmetrized Hausdorff Distance we can, WLOG, assume the maximum occurs with respect to z_1. So, we compute $\min_{1\leq j\leq n-1} |z_1 - w_j|$ iteratively by using the formula $min(a, b) = \frac{a+b-|a-b|}{2}$ and expressing $|a-b|$ by defining a new variable $y = |a-b|$, so that $y^2 - |a-b|^2 = 0$ and $y \geq 0$, as follows.

Notation 4.1.2.

$$x_{4n-1} = (x_1 - x_{2n+1})^2 + (x_2 - x_{2n+2})^2 = |z_1 - w_1|^2$$

If $n > 2$,

$$x_{4n+1} = \left|x_{4n-1} - (x_1 - x_{2n+3})^2 - (x_2 - x_{2n+4})^2\right| = \left||z_1 - w_1|^2 - |z_1 - w_2|^2\right|$$
$$x_{4n} = \frac{(x_1 - x_{2n+3})^2 + (x_2 - x_{2n+4})^2 + x_{4n-1} - x_{4n+1}}{2} = \min\{|z_1 - w_1|^2, |z_1 - w_2|^2\}$$

For $3 \le j \le n-1$

$$\begin{aligned} x_{4n+2(j-2)+1} &= \left|x_{4n+2(j-3)} - (x_1 - x_{2n+2j-1})^2 - (x_2 - x_{2n+2j})^2\right| \\ &= \left|\left(\min_{1\le \ell \le j-1} |z_1 - w_\ell|^2\right) - |z_1 - w_j|^2\right| \\ x_{4n+2(j-2)} &= \frac{(x_1 - x_{2n+2j-1})^2 + (x_2 - x_{2n+2j})^2 + x_{4n+2(j-3)} - x_{3n+2(j-2)+1}}{2} = \min_{1\le \ell \le j} |z_1 - w_j|^2 \end{aligned}$$

Thus, when $n > 2$,

$$x_{4n-6} = \min_{1\le j\le n-1} |z_1 - w_j|^2.$$

Semi-definite Constraints

One equality constriant:

$$x_{4n-1} - (x_1 - x_{2n+1})^2 - (x_2 - x_{2n+2})^2 = 0$$

If $n > 2$, One inequality constraint:

$$-x_{4n+1} \le 0$$

One equality constraint:

$$x_{4n+1}^2 - \left(x_{4n-1} - (x_1 - x_{2n+3})^2 - (x_2 - x_{2n+4})^2)\right)^2 = 0$$

Another equality constraint:

$$x_{4n} - \frac{(x_1 - x_{2n+3})^2 + (x_2 - x_{2n+4})^2 + x_{4n-1} - x_{4n+1}}{2} = 0$$

If $n > 3$, $n - 3$ inequality constraints:

$$-x_{4n+2(j-2)+1} \leq 0$$

$n - 3$ equality constraints:

$$x_{4n+2(j-2)+1}^2 - \left(x_{4n+2(j-3)} - (x_1 - x_{2n+2j-1})^2 - (x_2 - x_{2n+2j})^2\right)^2 = 0$$

Another $n - 3$ equality constraints:

$$x_{4n+2(j-2)} - \frac{(x_1 - x_{2n+2j-1})^2 + (x_2 - x_{2n+2j})^2 + x_{4n+2(j-3)} - x_{4n+2(j-2)+1}}{2} = 0$$

Respectively, the square of the unsymmetrized Hausdorff distance will be x_7 and x_{4n-6} when $n = 2$ and $n \geq 3$.

If $n = 2$, there are 5 equality constraints, two of which are redundant (so 3 unique equality constraints), and two inequality constraints. If $n = 3$, there are 11 unique equality constraints and and 4 inequality constraints. If $n \geq 4$, there are $6n - 7$ equality constraints [4n-4+1+2+2(n-3)] and $2n - 2$ inequality constraints [n+1+n-3].

What follows are the constraints for degrees two through four.

4.1.3 Degree 2

```
function [cIneq,cEq] =D2SCon(x)
cEq(1)=2*x(5) - x(3) - x(1);
cEq(2)=2*x(6) - x(4) - x(2);
cEq(3)=x(1) + x(3) - 2*x(5);
cEq(4)=x(2) + x(4) - 2*x(6);
cIneq(1)=x(1)^2+x(2)^2-1;
cIneq(2)=x(3)^2+x(4)^2-1;
cEq(5)=(x(1)-x(5))^2+(x(2)-x(6))^2-x(7);
```

4.1.4 Degree 3

```
function [cIneq,cEq] =D3SCon(x)
cEq(1)=x(1)*x(3) + x(1)*x(5) - x(2)*x(4) - 2*x(1)*x(7) - x
    (2)*x(6) + x(3)*x(5) + 2*x(2)*x(8) - 2*x(3)*x(7) - x(4)*
    x(6) + 2*x(4)*x(8) - 2*x(5)*x(7) + 2*x(6)*x(8) + 3*x(7)
    ^2 - 3*x(8)^2;
cEq(2)=x(1)*x(4) + x(2)*x(3) + x(1)*x(6) + x(2)*x(5) - 2*x
    (1)*x(8) - 2*x(2)*x(7) + x(3)*x(6) + x(4)*x(5) - 2*x(3)*
    x(8) - 2*x(4)*x(7) - 2*x(5)*x(8) - 2*x(6)*x(7) + 6*x(7)*
    x(8);
cEq(3)=x(1)*x(3) + x(1)*x(5) - x(2)*x(4) - x(2)*x(6) + x
    (3)*x(5) - 2*x(1)*x(9) - x(4)*x(6) + 2*x(2)*x(10) - 2*x
```

```
    (3)*x(9) + 2*x(4)*x(10) - 2*x(5)*x(9) + 2*x(6)*x(10) +
    3*x(9)^2 - 3*x(10)^2;
cEq(4)=x(1)*x(4) + x(2)*x(3) + x(1)*x(6) + x(2)*x(5) + x
    (3)*x(6) + x(4)*x(5) - 2*x(1)*x(10) - 2*x(2)*x(9) - 2*x
    (3)*x(10) - 2*x(4)*x(9) - 2*x(5)*x(10) - 2*x(6)*x(9) +
    6*x(9)*x(10);
cEq(5)=x(2)*x(4) - x(1)*x(5) - x(1)*x(3) + x(2)*x(6) - x
    (3)*x(5) + x(4)*x(6) + 3*x(7)*x(9) - 3*x(8)*x(10);
cEq(6)=3*x(7)*x(10) - x(2)*x(3) - x(1)*x(6) - x(2)*x(5) -
    x(3)*x(6) - x(4)*x(5) - x(1)*x(4) + 3*x(8)*x(9);
cEq(7)=2*x(1) + 2*x(3) + 2*x(5) - 3*x(7) - 3*x(9);
cEq(8)=2*x(2) + 2*x(4) + 2*x(6) - 3*x(8) - 3*x(10);
cIneq(1)=x(1)^2+x(2)^2-1;
cIneq(2)=x(3)^2+x(4)^2-1;
cIneq(3)=x(5)^2+x(6)^2-1;
cEq(9)=(x(1)-x(7))^2+(x(2)-x(8))^2-x(11);
cIneq(4)= -x(13);
cEq(10)=x(13)^2-(x(11)-(x(1)-x(9))^2-(x(2)-x(10))^2)^2;
cEq(11)=x(12)-((x(1)-x(9))^2+(x(2)-x(10))^2+x(11)-x(13))
    /2;
```

4.1.5 Degree 4

```
function [cIneq,cEq] =D4SCon(x)
cEq(1)=3*x(1)*x(10)^2 - 3*x(1)*x(9)^2 - 3*x(3)*x(9)^2 + 3*
    x(3)*x(10)^2 - 3*x(5)*x(9)^2 + 3*x(5)*x(10)^2 - 3*x(7)*x
    (9)^2 + 3*x(7)*x(10)^2 - 12*x(9)*x(10)^2 + 4*x(9)^3 - x
    (1)*x(3)*x(5) - x(1)*x(3)*x(7) + x(1)*x(4)*x(6) + x(2)*x
    (3)*x(6) + x(2)*x(4)*x(5) + 2*x(1)*x(3)*x(9) + x(1)*x(4)
    *x(8) - x(1)*x(5)*x(7) + x(2)*x(3)*x(8) + x(2)*x(4)*x(7)
     - 2*x(1)*x(4)*x(10) + 2*x(1)*x(5)*x(9) + x(1)*x(6)*x(8)
     - 2*x(2)*x(3)*x(10) - 2*x(2)*x(4)*x(9) + x(2)*x(5)*x(8)
     + x(2)*x(6)*x(7) - x(3)*x(5)*x(7) - 2*x(1)*x(6)*x(10) +
     2*x(1)*x(7)*x(9) - 2*x(2)*x(5)*x(10) - 2*x(2)*x(6)*x(9)
     + 2*x(3)*x(5)*x(9) + x(3)*x(6)*x(8) + x(4)*x(5)*x(8) +
    x(4)*x(6)*x(7) - 2*x(1)*x(8)*x(10) - 2*x(2)*x(7)*x(10) -
     2*x(2)*x(8)*x(9) - 2*x(3)*x(6)*x(10) + 2*x(3)*x(7)*x(9)
     - 2*x(4)*x(5)*x(10) - 2*x(4)*x(6)*x(9) + 6*x(2)*x(9)*x
    (10) - 2*x(3)*x(8)*x(10) - 2*x(4)*x(7)*x(10) - 2*x(4)*x
    (8)*x(9) + 2*x(5)*x(7)*x(9) + 6*x(4)*x(9)*x(10) - 2*x(5)
    *x(8)*x(10) - 2*x(6)*x(7)*x(10) - 2*x(6)*x(8)*x(9) + 6*x
    (6)*x(9)*x(10) + 6*x(8)*x(9)*x(10);
```

```
cEq(2)=3*x(2)*x(10)^2 - 3*x(2)*x(9)^2 - 3*x(4)*x(9)^2 + 3*
    x(4)*x(10)^2 - 3*x(6)*x(9)^2 + 3*x(6)*x(10)^2 - 3*x(8)*x
    (9)^2 + 3*x(8)*x(10)^2 + 12*x(9)^2*x(10) - 4*x(10)^3 - x
    (1)*x(3)*x(6) - x(1)*x(4)*x(5) - x(2)*x(3)*x(5) - x(1)*x
    (3)*x(8) - x(1)*x(4)*x(7) - x(2)*x(3)*x(7) + x(2)*x(4)*x
    (6) + 2*x(1)*x(3)*x(10) + 2*x(1)*x(4)*x(9) - x(1)*x(5)*x
    (8) - x(1)*x(6)*x(7) + 2*x(2)*x(3)*x(9) + x(2)*x(4)*x(8)
     - x(2)*x(5)*x(7) + 2*x(1)*x(5)*x(10) + 2*x(1)*x(6)*x(9)
     - 2*x(2)*x(4)*x(10) + 2*x(2)*x(5)*x(9) + x(2)*x(6)*x(8)
     - x(3)*x(5)*x(8) - x(3)*x(6)*x(7) - x(4)*x(5)*x(7) + 2*
    x(1)*x(7)*x(10) + 2*x(1)*x(8)*x(9) - 2*x(2)*x(6)*x(10) +
     2*x(2)*x(7)*x(9) + 2*x(3)*x(5)*x(10) + 2*x(3)*x(6)*x(9)
     + 2*x(4)*x(5)*x(9) + x(4)*x(6)*x(8) - 6*x(1)*x(9)*x(10)
     - 2*x(2)*x(8)*x(10) + 2*x(3)*x(7)*x(10) + 2*x(3)*x(8)*x
    (9) - 2*x(4)*x(6)*x(10) + 2*x(4)*x(7)*x(9) - 6*x(3)*x(9)
    *x(10) - 2*x(4)*x(8)*x(10) + 2*x(5)*x(7)*x(10) + 2*x(5)*
    x(8)*x(9) + 2*x(6)*x(7)*x(9) - 6*x(5)*x(9)*x(10) - 2*x
    (6)*x(8)*x(10) - 6*x(7)*x(9)*x(10);
cEq(3)=3*x(1)*x(12)^2 - 3*x(1)*x(11)^2 - 3*x(3)*x(11)^2 +
    3*x(3)*x(12)^2 - 3*x(5)*x(11)^2 + 3*x(5)*x(12)^2 - 3*x
    (7)*x(11)^2 + 3*x(7)*x(12)^2 - 12*x(11)*x(12)^2 + 4*x
```

```
(11)^3 - x(1)*x(3)*x(5) - x(1)*x(3)*x(7) + x(1)*x(4)*x
(6) + x(2)*x(3)*x(6) + x(2)*x(4)*x(5) + x(1)*x(4)*x(8) -
 x(1)*x(5)*x(7) + x(2)*x(3)*x(8) + x(2)*x(4)*x(7) + 2*x
(1)*x(3)*x(11) + x(1)*x(6)*x(8) + x(2)*x(5)*x(8) + x(2)*
x(6)*x(7) - x(3)*x(5)*x(7) - 2*x(1)*x(4)*x(12) + 2*x(1)*
x(5)*x(11) - 2*x(2)*x(3)*x(12) - 2*x(2)*x(4)*x(11) + x
(3)*x(6)*x(8) + x(4)*x(5)*x(8) + x(4)*x(6)*x(7) - 2*x(1)
*x(6)*x(12) + 2*x(1)*x(7)*x(11) - 2*x(2)*x(5)*x(12) - 2*
x(2)*x(6)*x(11) + 2*x(3)*x(5)*x(11) - 2*x(1)*x(8)*x(12)
- 2*x(2)*x(7)*x(12) - 2*x(2)*x(8)*x(11) - 2*x(3)*x(6)*x
(12) + 2*x(3)*x(7)*x(11) - 2*x(4)*x(5)*x(12) - 2*x(4)*x
(6)*x(11) - 2*x(3)*x(8)*x(12) - 2*x(4)*x(7)*x(12) - 2*x
(4)*x(8)*x(11) + 2*x(5)*x(7)*x(11) + 6*x(2)*x(11)*x(12)
- 2*x(5)*x(8)*x(12) - 2*x(6)*x(7)*x(12) - 2*x(6)*x(8)*x
(11) + 6*x(4)*x(11)*x(12) + 6*x(6)*x(11)*x(12) + 6*x(8)*
x(11)*x(12);
cEq(4)=3*x(2)*x(12)^2 - 3*x(2)*x(11)^2 - 3*x(4)*x(11)^2 +
3*x(4)*x(12)^2 - 3*x(6)*x(11)^2 + 3*x(6)*x(12)^2 - 3*x
(8)*x(11)^2 + 3*x(8)*x(12)^2 + 12*x(11)^2*x(12) - 4*x
(12)^3 - x(1)*x(3)*x(6) - x(1)*x(4)*x(5) - x(2)*x(3)*x
(5) - x(1)*x(3)*x(8) - x(1)*x(4)*x(7) - x(2)*x(3)*x(7) +
```

36

```
 x(2)*x(4)*x(6) - x(1)*x(5)*x(8) - x(1)*x(6)*x(7) + x(2)
*x(4)*x(8) - x(2)*x(5)*x(7) + 2*x(1)*x(3)*x(12) + 2*x(1)
*x(4)*x(11) + 2*x(2)*x(3)*x(11) + x(2)*x(6)*x(8) - x(3)*
x(5)*x(8) - x(3)*x(6)*x(7) - x(4)*x(5)*x(7) + 2*x(1)*x
(5)*x(12) + 2*x(1)*x(6)*x(11) - 2*x(2)*x(4)*x(12) + 2*x
(2)*x(5)*x(11) + x(4)*x(6)*x(8) + 2*x(1)*x(7)*x(12) + 2*
x(1)*x(8)*x(11) - 2*x(2)*x(6)*x(12) + 2*x(2)*x(7)*x(11)
+ 2*x(3)*x(5)*x(12) + 2*x(3)*x(6)*x(11) + 2*x(4)*x(5)*x
(11) - 2*x(2)*x(8)*x(12) + 2*x(3)*x(7)*x(12) + 2*x(3)*x
(8)*x(11) - 2*x(4)*x(6)*x(12) + 2*x(4)*x(7)*x(11) - 6*x
(1)*x(11)*x(12) - 2*x(4)*x(8)*x(12) + 2*x(5)*x(7)*x(12)
+ 2*x(5)*x(8)*x(11) + 2*x(6)*x(7)*x(11) - 6*x(3)*x(11)*x
(12) - 2*x(6)*x(8)*x(12) - 6*x(5)*x(11)*x(12) - 6*x(7)*x
(11)*x(12);
cEq(5)=3*x(1)*x(14)^2 - 3*x(1)*x(13)^2 - 3*x(3)*x(13)^2 +
3*x(3)*x(14)^2 - 3*x(5)*x(13)^2 + 3*x(5)*x(14)^2 - 3*x
(7)*x(13)^2 + 3*x(7)*x(14)^2 - 12*x(13)*x(14)^2 + 4*x
(13)^3 - x(1)*x(3)*x(5) - x(1)*x(3)*x(7) + x(1)*x(4)*x
(6) + x(2)*x(3)*x(6) + x(2)*x(4)*x(5) + x(1)*x(4)*x(8) -
 x(1)*x(5)*x(7) + x(2)*x(3)*x(8) + x(2)*x(4)*x(7) + x(1)
*x(6)*x(8) + x(2)*x(5)*x(8) + x(2)*x(6)*x(7) - x(3)*x(5)
```

```
*x(7) + 2*x(1)*x(3)*x(13) + x(3)*x(6)*x(8) + x(4)*x(5)*x
(8) + x(4)*x(6)*x(7) - 2*x(1)*x(4)*x(14) + 2*x(1)*x(5)*x
(13) - 2*x(2)*x(3)*x(14) - 2*x(2)*x(4)*x(13) - 2*x(1)*x
(6)*x(14) + 2*x(1)*x(7)*x(13) - 2*x(2)*x(5)*x(14) - 2*x
(2)*x(6)*x(13) + 2*x(3)*x(5)*x(13) - 2*x(1)*x(8)*x(14) -
 2*x(2)*x(7)*x(14) - 2*x(2)*x(8)*x(13) - 2*x(3)*x(6)*x
(14) + 2*x(3)*x(7)*x(13) - 2*x(4)*x(5)*x(14) - 2*x(4)*x
(6)*x(13) - 2*x(3)*x(8)*x(14) - 2*x(4)*x(7)*x(14) - 2*x
(4)*x(8)*x(13) + 2*x(5)*x(7)*x(13) - 2*x(5)*x(8)*x(14) -
 2*x(6)*x(7)*x(14) - 2*x(6)*x(8)*x(13) + 6*x(2)*x(13)*x
(14) + 6*x(4)*x(13)*x(14) + 6*x(6)*x(13)*x(14) + 6*x(8)*
x(13)*x(14);
cEq(6)=3*x(2)*x(14)^2 - 3*x(2)*x(13)^2 - 3*x(4)*x(13)^2 +
 3*x(4)*x(14)^2 - 3*x(6)*x(13)^2 + 3*x(6)*x(14)^2 - 3*x
(8)*x(13)^2 + 3*x(8)*x(14)^2 + 12*x(13)^2*x(14) - 4*x
(14)^3 - x(1)*x(3)*x(6) - x(1)*x(4)*x(5) - x(2)*x(3)*x
(5) - x(1)*x(3)*x(8) - x(1)*x(4)*x(7) - x(2)*x(3)*x(7) +
 x(2)*x(4)*x(6) - x(1)*x(5)*x(8) - x(1)*x(6)*x(7) + x(2)
*x(4)*x(8) - x(2)*x(5)*x(7) + x(2)*x(6)*x(8) - x(3)*x(5)
*x(8) - x(3)*x(6)*x(7) - x(4)*x(5)*x(7) + 2*x(1)*x(3)*x
(14) + 2*x(1)*x(4)*x(13) + 2*x(2)*x(3)*x(13) + x(4)*x(6)
```

38

```
    *x(8) + 2*x(1)*x(5)*x(14) + 2*x(1)*x(6)*x(13) - 2*x(2)*x
    (4)*x(14) + 2*x(2)*x(5)*x(13) + 2*x(1)*x(7)*x(14) + 2*x
    (1)*x(8)*x(13) - 2*x(2)*x(6)*x(14) + 2*x(2)*x(7)*x(13) +
     2*x(3)*x(5)*x(14) + 2*x(3)*x(6)*x(13) + 2*x(4)*x(5)*x
    (13) - 2*x(2)*x(8)*x(14) + 2*x(3)*x(7)*x(14) + 2*x(3)*x
    (8)*x(13) - 2*x(4)*x(6)*x(14) + 2*x(4)*x(7)*x(13) - 2*x
    (4)*x(8)*x(14) + 2*x(5)*x(7)*x(14) + 2*x(5)*x(8)*x(13) +
     2*x(6)*x(7)*x(13) - 6*x(1)*x(13)*x(14) - 2*x(6)*x(8)*x
    (14) - 6*x(3)*x(13)*x(14) - 6*x(5)*x(13)*x(14) - 6*x(7)*
    x(13)*x(14);
cEq(7)=x(1)*x(3)*x(5) + x(1)*x(3)*x(7) - x(1)*x(4)*x(6) -
    x(2)*x(3)*x(6) - x(2)*x(4)*x(5) - x(1)*x(4)*x(8) + x(1)*
    x(5)*x(7) - x(2)*x(3)*x(8) - x(2)*x(4)*x(7) - x(1)*x(6)*
    x(8) - x(2)*x(5)*x(8) - x(2)*x(6)*x(7) + x(3)*x(5)*x(7)
    - x(3)*x(6)*x(8) - x(4)*x(5)*x(8) - x(4)*x(6)*x(7) - 4*x
    (9)*x(11)*x(13) + 4*x(9)*x(12)*x(14) + 4*x(10)*x(11)*x
    (14) + 4*x(10)*x(12)*x(13);
cEq(8)=x(1)*x(3)*x(6) + x(1)*x(4)*x(5) + x(2)*x(3)*x(5) +
    x(1)*x(3)*x(8) + x(1)*x(4)*x(7) + x(2)*x(3)*x(7) - x(2)*
    x(4)*x(6) + x(1)*x(5)*x(8) + x(1)*x(6)*x(7) - x(2)*x(4)*
    x(8) + x(2)*x(5)*x(7) - x(2)*x(6)*x(8) + x(3)*x(5)*x(8)
```

39

```
    + x(3)*x(6)*x(7) + x(4)*x(5)*x(7) - x(4)*x(6)*x(8) - 4*x
    (9)*x(11)*x(14) - 4*x(9)*x(12)*x(13) - 4*x(10)*x(11)*x
    (13) + 4*x(10)*x(12)*x(14);
cEq(9)=2*x(2)*x(4) - 2*x(1)*x(5) - 2*x(1)*x(3) - 2*x(1)*x
    (7) + 2*x(2)*x(6) - 2*x(3)*x(5) + 2*x(2)*x(8) - 2*x(3)*x
    (7) + 2*x(4)*x(6) + 2*x(4)*x(8) - 2*x(5)*x(7) + 2*x(6)*x
    (8) + 4*x(9)*x(11) + 4*x(9)*x(13) - 4*x(10)*x(12) - 4*x
    (10)*x(14) + 4*x(11)*x(13) - 4*x(12)*x(14);
cEq(10)=4*x(9)*x(12) - 2*x(2)*x(3) - 2*x(1)*x(6) - 2*x(2)*
    x(5) - 2*x(1)*x(8) - 2*x(2)*x(7) - 2*x(3)*x(6) - 2*x(4)*
    x(5) - 2*x(3)*x(8) - 2*x(4)*x(7) - 2*x(5)*x(8) - 2*x(6)*
    x(7) - 2*x(1)*x(4) + 4*x(10)*x(11) + 4*x(9)*x(14) + 4*x
    (10)*x(13) + 4*x(11)*x(14) + 4*x(12)*x(13);
cEq(11)=3*x(1) + 3*x(3) + 3*x(5) + 3*x(7) - 4*x(9) - 4*x
    (11) - 4*x(13);
cEq(12)=3*x(2) + 3*x(4) + 3*x(6) + 3*x(8) - 4*x(10) - 4*x
    (12) - 4*x(14);
cIneq(1)=x(1)^2+x(2)^2-1;
cIneq(2)=x(3)^2+x(4)^2-1;
cIneq(3)=x(5)^2+x(6)^2-1;
cIneq(4)=x(7)^2+x(8)^2-1;
```

```
cEq(13)=(x(1)-x(9))^2+(x(2)-x(10))^2-x(15);
cIneq(5)= -x(17);
cEq(14)=x(17)^2-(x(15)-(x(1)-x(11))^2-(x(2)-x(12))^2)^2;
cEq(15)=x(16)-((x(1)-x(11))^2+(x(2)-x(12))^2+x(15)-x(17))
    /2;
cIneq(6)=-x(19);
cEq(16)=x(19)^2-(x(16)-(x(1)-x(13))^2-(x(2)-x(14))^2)^2;
cEq(17)=x(18)-((x(1)-x(13))^2+(x(2)-x(14))^2+x(16)-x(19))
    /2;
```

4.2 Local Minimization

Remark 4.2.1. The primary purpose of this section is to verify the optimization constraints and initialization procedures are working. We do this by confirming the roots of unity are local minimal. We also check the local minima found by Miller in degrees 6 and 8 with the same input as in §2.5.2.

4.2.1 Verifying Roots of Unity Are Local Minimum

Remark 4.2.2. During verification that the roots of unity are local minima, the optimization procedure results in better precision than the companion matrix method. This is in part because, in the case of the roots of unity, the companion matrix method is ill-conditioned. Furthermore, the optimization procedure uses

n	CLIE	OLIE
2	6×10^{-17}	0
3	3×10^{-6}	3×10^{-14}
4	4×10^{-6}	0
5	3×10^{-3}	6×10^{-8}
6	3×10^{-3}	1×10^{-10}
7	1×10^{-2}	2×10^{-15}
8	1×10^{-2}	3×10^{-11}
9	4×10^{-2}	3×10^{-11}

Figure 4.1: Error Comparison in Computation of Critical Points of Roots of Unity between Companion Matrix Method versus Local Optimization Using Moment Constraints

the companion matrix technique in initialization, and uses orders of magnitude more FLOPS and runtime. Regardless, the increase in precision is remarkable. Recall that the critical points of $f(z) = z^n - 1$ will be zero with multiplicity $n-1$. Call the maximum of the absolute values of the computed critical points using the optimization procedure the 'optimization ℓ^∞ error,' OLIE. Call the maximum of the absolute values of the computed critical points using the companion matrix technique the 'companion ℓ^∞ error,' CLIE. Those results are summarized in Figure 4.1, wherein 0 indicates zero within machine double precision.

4.2.2 Verify Miller's Counterexamples are Local Minima

General Sendov Conjecture Local Extrema Searcher

```
function GenDegSCESFMCAWRTDSWRTZL(n)
%stands for
```

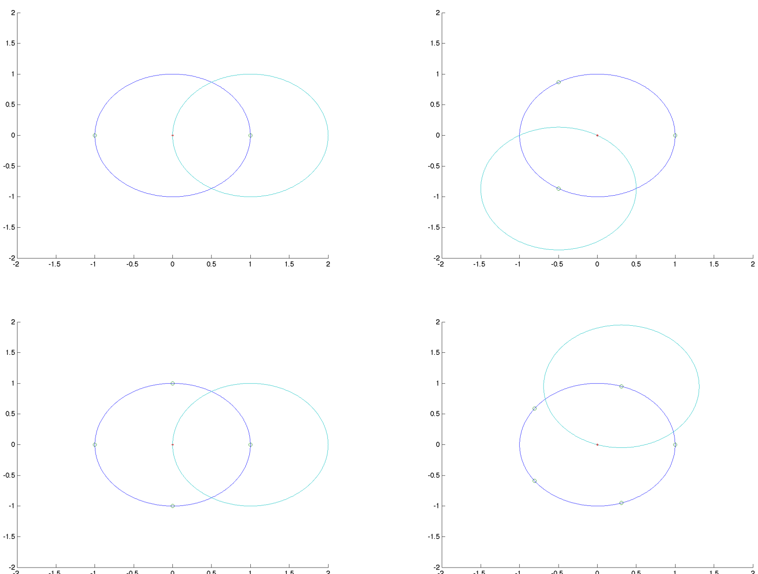


Figure 4.2: Roots of Unity, Fmincon Degrees 2-5

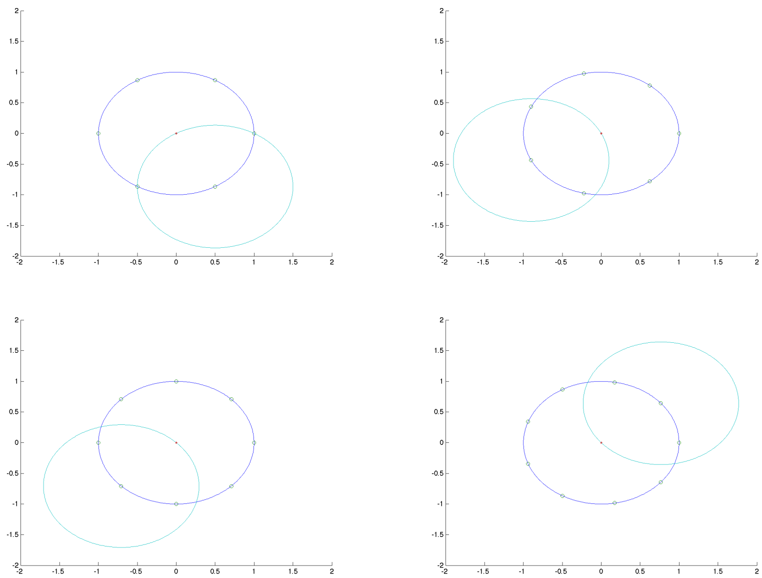


Figure 4.3: Roots of Unity, Fmincon Degrees 6-9

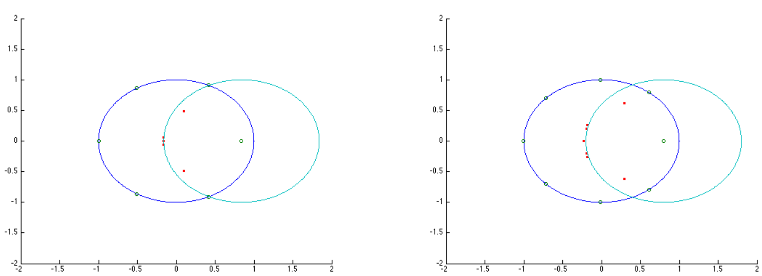


Figure 4.4: Miller's Examples are Local Minima In Degrees 6 and 8.

44

```
%General Degree Sendov Conjecture Extrema Searcher using
    fmincon, Asymmetric With Respect To Hausdorff Distance,
    Symmetric With Respect To Zero Location
        N=6*n-5;

        combinedFilename = strcat('Deg',int2str(n),'
            SendovConjectureExtremaSearcher.m');
        %Degree n Borcea p-Variance Objective File
        combinedObjectiveFilename = strcat('D',int2str(n),
            'SObj.m');
        %Degree n Borcea p-Variance Constraint File
        combinedConstraintFilename = strcat('D',int2str(n)
            ,'SCon.m');

        clearsyms='clear global q h';
        symInstantiation='syms ';
        for k=1:N
                clearsyms=strcat(clearsyms,' y',int2str(k)
                    );
                symInstantiation=strcat(symInstantiation,'
                    y',int2str(k));
```

```
end
symInstantiation=strcat(symInstantiation,' real');
syms eta;
eval(clearsyms);
eval(symInstantiation);

f3 = fopen(combinedConstraintFilename,'w');
helperInstantiation = strcat('function [cIneq,cEq]
    = ',combinedConstraintFilename(1:length(
    combinedConstraintFilename)-2),'(x)\n');
fprintf(f3,helperInstantiation);
cEqConstraintIndex=1;
cIneqConstraintIndex=1;

%Constraints which define critical points.
display('Using symbolic toolbox to generate
    constraints which define critical points.');
for j=1:n-1
        h=0;
        for k=1:n
                stringEquation='1';
```

```
        for l=1:n
            if (l~=k)
                stringEquation=
                    strcat(
                    stringEquation,'
                    *((y',int2str(2*
                    n+2*j-1),'-y',
                    int2str(2*l-1),'
                    )+(y',int2str(2*
                    n+2*j),'-y',
                    int2str(2*l),')*
                    i)');
            end
        end
        q=expand(eval(stringEquation));
        h=q+h;
    end

    helperInstantiation =
        GenDegStaticStringReplacerHelper(char(
        eval(real(h))));
```

47

```
        helperInstantiation = strcat('cEq(',
            int2str(cEqConstraintIndex),')=',
            helperInstantiation,';\n');
        fprintf(f3,helperInstantiation);
        cEqConstraintIndex = cEqConstraintIndex+1;

        helperInstantiation =
            GenDegStaticStringReplacerHelper(char(
            eval(imag(h))));
        helperInstantiation = strcat('cEq(',
            int2str(cEqConstraintIndex),')=',
            helperInstantiation,';\n');
        fprintf(f3,helperInstantiation);
        cEqConstraintIndex = cEqConstraintIndex+1;
end

stringEquation='1';
for k=1:n
        stringEquation=strcat(stringEquation,'*(
            eta -(y',int2str(2*k-1),'+y',int2str(2*k
            ),'*i))');
```

48

```
end
q=expand(diff(eval(stringEquation),eta));

stringEquation='1';
for j=1:n-1
        stringEquation =strcat(stringEquation,'*(
            eta-(y',int2str(2*n+2*j-1),'+y',int2str
            (2*n+2*j),'*i))');
end
h=expand(n*eval(stringEquation));

momentRestrictions = coeffs(h-q,eta);

display(momentRestrictions);

display('Using symbolic toolbox to generate 
    contraints based on moments of Critical Points')
    ;
for ell=1:n-1
```

49

```
            helperInstantiation =
               GenDegStaticStringReplacerHelper(char(
               eval(real(momentRestrictions(ell)))));
            helperInstantiation = strcat('cEq(',
               int2str(cEqConstraintIndex),')=',
               helperInstantiation,';\n');
            fprintf(f3,helperInstantiation);
            cEqConstraintIndex = cEqConstraintIndex+1;

            helperInstantiation =
               GenDegStaticStringReplacerHelper(char(
               eval(imag(momentRestrictions(ell)))));
            helperInstantiation = strcat('cEq(',
               int2str(cEqConstraintIndex),')=',
               helperInstantiation,';\n');
            fprintf(f3,helperInstantiation);
            cEqConstraintIndex = cEqConstraintIndex+1;
end

display('Generating constraints which ensure zeros
    are contained withing closed unit disk');
```

```
for k=1:n
        helperInstantiation = strcat('x(',int2str
            (2*k-1),')^2+x(',int2str(2*k),')^2-1');
        helperInstantiation = strcat('cIneq(',
            int2str(cIneqConstraintIndex),')=',
            helperInstantiation,';\n');
        fprintf(f3,helperInstantiation);
        cIneqConstraintIndex =
            cIneqConstraintIndex+1;
end

%Constraints which define minimum and maximum. Use
     for computation of unsymmetrized Hausdorff
    distance
display('Generating constraints which are used in
    computation of unsymmetrized Hausdorf distance')
    ;

helperInstantiation = strcat('(x(1)-x(',int2str(2*
    n+1),'))^2+(x(2)-x(',int2str(2*n+2),'))^2-x(',
    int2str(4*n-1),')');
```

```
helperInstantiation = strcat('cEq(',int2str(
   cEqConstraintIndex),')=',helperInstantiation,';\
   n');
fprintf(f3,helperInstantiation);
cEqConstraintIndex = cEqConstraintIndex+1;

%j=2 follows
if(n >2)
        helperInstantiation = strcat('cIneq(',
           int2str(cIneqConstraintIndex),')= -x(',
           int2str(4*n+1),');\n');
        fprintf(f3,helperInstantiation);
        cIneqConstraintIndex =
           cIneqConstraintIndex+1;

        helperInstantiation = strcat('x(',int2str
           (4*n+1),')^2-(x(',int2str(4*n-1),')-(x
           (1)-x(',int2str(2*n+3),'))^2-(x(2)-x(',
           int2str(2*n+4),'))^2)^2');
```

```
helperInstantiation = strcat('cEq(',
   int2str(cEqConstraintIndex),')=',
   helperInstantiation,';\n');
fprintf(f3,helperInstantiation);
cEqConstraintIndex = cEqConstraintIndex+1;

helperInstantiation = strcat('x(',int2str
   (4*n),')-((x(1)-x(',int2str(2*n+3),'))
   ^2+(x(2)-x(',int2str(2*n+4),'))^2+x(',
   int2str(4*n-1),')-x(',int2str(4*n+1),'))
   /2');
helperInstantiation = strcat('cEq(',
   int2str(cEqConstraintIndex),')=',
   helperInstantiation,';\n');
fprintf(f3,helperInstantiation);
cEqConstraintIndex = cEqConstraintIndex+1;

for j=3:n-1
      helperInstantiation = strcat('
         cIneq(',int2str(
```

53

```
    cIneqConstraintIndex),')=-x(',
    int2str(4*n+2*(j-2)+1),');\n');
fprintf(f3,helperInstantiation);
cIneqConstraintIndex =
    cIneqConstraintIndex+1;

helperInstantiation = strcat('x(',
    int2str(4*n+2*(j-2)+1),')^2-(x('
    ,int2str(4*n+2*(j-3)),')-(x(1)-x
    (',int2str(2*n+2*j-1),'))^2-(x
    (2)-x(',int2str(2*n+2*j),'))^2)
    ^2');
helperInstantiation = strcat('cEq(
    ',int2str(cEqConstraintIndex),')
    =',helperInstantiation,';\n');
fprintf(f3,helperInstantiation);
cEqConstraintIndex =
    cEqConstraintIndex+1;

helperInstantiation = strcat('x(',
    int2str(4*n+2*(j-2)),')-((x(1)-x
```

54

```
                    (',int2str(2*n+2*j-1),'))^2+(x
                    (2)-x(',int2str(2*n+2*j),'))^2+x
                    (',int2str(4*n+2*(j-3)),')-x(',
                    int2str(4*n+2*(j-2)+1),'))/2');
                helperInstantiation = strcat('cEq(
                    ',int2str(cEqConstraintIndex),')
                    =',helperInstantiation,';\n');
                fprintf(f3,helperInstantiation);
                cEqConstraintIndex =
                    cEqConstraintIndex+1;
        end
end

fclose(f3);

f2 = fopen(combinedObjectiveFilename,'w');
helperInstantiation = strcat('function f = ',
    combinedObjectiveFilename(1:length(
    combinedObjectiveFilename)-2),'(x)\n');
fprintf(f2,helperInstantiation);
```

```
if(n==2)
        helperInstantiation = strcat('f=-x(7)');
else
        helperInstantiation = strcat('f=-x(',
            int2str(6*n-6),')');
end
fprintf(f2,helperInstantiation);

fclose(f2);

fid = fopen(combinedFilename,'w');

initialCondition = 'x0 = [1,0';
for k=2:n
        initialCondition = strcat(initialCondition
            ,',',num2str(real(exp(2*(k-1)*pi*i/n))),
            ',',num2str(imag(exp(2*(k-1)*pi*i/n))));
end
for j=1:n-1
        initialCondition = strcat(initialCondition
            ,',0,0');
```

```
end
initialCondition = strcat(initialCondition,',1');
for ell=2*n+1:N
        initialCondition = strcat(initialCondition
            ,',1,0');
end
initialCondition = strcat(initialCondition,'];\n')
    ;

fprintf(fid,initialCondition);

helperInstantiation = 'options = optimset(''
    Algorithm'',''active-set'',''Display'',''iter'')
    ;\n';
fprintf(fid,helperInstantiation);

fprintf(fid,'\n');
helperInstantiation = strcat('[xStar,fval,exitflag
    ,output,lambda,grad,hessian] =fmincon(@',
    combinedObjectiveFilename(1:length(
    combinedObjectiveFilename)-2),',x0
```

```
    ,[] ,[] ,[] ,[] ,[] ,[] ,@',combinedConstraintFilename
    (1:length(combinedConstraintFilename)-2),',
    options);\n');
fprintf(fid,helperInstantiation);

interpretResults = 'z=[';
for k=1:n
        interpretResults = strcat(interpretResults
            ,'double(xStar(',int2str(2*k-1),'))+i*
            double(xStar(',int2str(2*k),')),');
end
interpretResults=strcat(interpretResults(1:length(
    interpretResults)-1),'];\n');
fprintf(fid,interpretResults);

interpretResults = 'w=[';
for j=1:n-1
        interpretResults = strcat(interpretResults
            ,'double(xStar(',int2str(2*n+2*j-1),'))+
            i*double(xStar(',int2str(2*n+2*j)),')),')
            ;
```

```
end
interpretResults=strcat(interpretResults(1:length(
    interpretResults)-1),'];\n');
fprintf(fid,interpretResults);


interpretResults = strcat('save(''',
    combinedFilename(1:length(combinedFilename)-2),'
    '');');
fprintf(fid,interpretResults);


fprintf(fid,'\n');
interpretResults = 'close(gcf);\n';
fprintf(fid,interpretResults);
interpretResults = 'BorceaCompanionMax(z);\n';
fprintf(fid,interpretResults);
interpretResults = strcat('saveas(gcf,''',
    combinedFilename(1:length(combinedFilename)-1),'
    pdf'',''pdf'');\n');
fprintf(fid,interpretResults);


fclose(fid);
```

4.3 YALMIP

Using the constraints verified for the local minimization procedure, we apply YALMIP's BMIBNB to attempt global optimization. Within machine error, the results is one of the conjectured optimal polynomials [c.f. §2 Conjecture 2 , pg. 4 of [**?**]].

To format the constraints used by Fmincon into a form YALMIP can use, we run the following code in a unix environment:

```
sed 's/cEq([0-9]\+)/[K,0=/g' D2SCon.m |
sed 's/cIneq([0-9]\+)/[K,0>/g' |
sed 's/;/];/g' |
sed '/^function/d' > D2SYCon.m
```

Let $c, z_k, w_j \in \mathbb{C}$, $1 \leq k \leq n$, $1 \leq j \leq n-1$. $z_k = x_{2k-1} + x_{2k}i$ $w_j = x_{2n+2j-1} + x_{2n+2j}i$

With constraint, for each j,

$$\sum_k \frac{1}{w_j - z_k} = 0 \Leftrightarrow 0 = \sum_{k=1}^{n} \prod_{\ell=1, \ell \neq k}^{n} (w_j - z_\ell)$$

So that $p(z) = \alpha \prod_{k=1}^{n}(z - z_k)$ and $p'(z) = \beta \prod_{k=1}^{n-1}(w_j - z_\ell)$. Our goal is to seek counterexamples of, or find further numerical evidence for, Borcea's 2-variance conjecture (pg. 4 Khavinson, Pereira, Putinar, Saff, and Shimorin), for each n.

In other words, we seek to minimize

$$(\sigma_2(p))^2 - (h(p,p'))^2.$$

If the above is non-negative for an extremal polynomial, further numerical evidence has been found. If the above is negative for an extremal polynomial, a counterexample to Borcea's 2-variance conjecture has been found.

Along the way, we similarly seek counterexamples of, or further numerical evidence for, Sendov's conjecture.

Where

$$h(p,p') := max_{p(z)=0} min_{p'(w)=0} |z-w|,$$

so that

$$(h(p,p'))^2 := max_{p(z)=0} min_{p'(w)=0} |z-w|^2,$$

and

$$(\sigma_2(p))^2 = min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2 \right).$$

Where

$$c = \frac{1}{n} \sum_{k=1}^{n} z_k.$$

We want to express $\max_k \min_j |z_k - w_j|^2$ as a polynomial, so that the optimization problem

$$min_{c,z_k,w_j} \left(min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2 \right) - \max_k \min_j |z_k - w_j|^2 \right)$$

subject to j contraints

$$0 = \sum_{k=1}^{n} \prod_{\ell=1,\ell\neq k}^{n} (w_j - z_\ell)$$

is one that can be solved using gloptipoly3 or YALMIP.

We recursively express the minimum for fixed k, by using the formula

$$min(a,b) = \frac{a+b-|a-b|}{2}$$

and expressing $|a-b|$ by defining a new variable $y = |a-b|$, so that $y^2 - |a-b|^2 = 0$ and $y \geq 0$.

And we express the maximum as

$$max(c,d) = \frac{c+d+|c-d|}{2}$$

with $z = |c-d|$, $z^2 - |c-d|^2 = 0$ and $z \geq 0$.

4.3.1 Degree 2 Example

$n = 2$

$$BasicVariables$$

$$c = x_1 + x_2 i$$

$$z_1 = x_3 + x_4 i$$

$$z_2 = x_5 + x_6 i$$

$$w_1 = x_7 + x_8 i$$

$$Constraint$$

$$(w_1 - z_2) + (w_1 - z_1) = 0$$

$$2x_7 - x_3 - x_5 = 0$$

$$2x_8 - x_4 - x_6 = 0$$

$$Variables and Constraints for Computing minimum$$

$$NONE(there is only one w_j)$$

$$Variables and Constraints for Computing maximum$$

$$a = |z_1 - w_1|^2 = (x_3 - x_7)^2 + (x_4 - x_8)^2$$

$$b = |z_2 - w_1|^2 = (x_5 - x_7)^2 + (x_6 - x_8)^2$$

$$x_9 = |a - b|$$

$$x_9^2 = (((x_3 - x_7)^2 + (x_4 - x_8)^2) - ((x_5 - x_7)^2 + (x_6 - x_8)^2))^2$$

$$x_9 \geq 0$$

Objective

$$|z_1 - c|^2 = |(x_3 - x_1) + (x_4 - x_2)i|^2 = (x_3 - x_1)^2 + (x_4 - x_2)^2$$

$$|z_2 - c|^2 = (x_5 - x_1)^2 + (x_6 - x_2)^2$$

$$min \left(\frac{(x_3 - x_1)^2 + (x_4 - x_2)^2 + (x_5 - x_1)^2 + (x_6 - x_2)^2}{2} - \frac{(x_3 - x_7)^2 + (x_4 - x_8)^2 + (x_5 - x_7)^2 + (x_6 - x_8)^2 + x_9}{2} \right)$$

```
sdpvar x1 x2 x3 x4 x5 x6 x7 x8 x9

F = [2*x7-x3-x5==0,2*x8-x4-x6==0,x9^2-(((x3-x7)^2+(x4-x8)^2)-((x5-x7)^2+(x6-x8)

F = [F, -100 < [x1 x2 x3 x4 x5 x6 x7 x8 x9] < 100];

options=sdpsettings('verbose',1,'solver','bmibnb');

solvesdp(F,(((x3-x1)^2+(x4-x2)^2+(x5-x1)^2+(x6-x2)^2)-((x3-x7)^2+(x4-x8)^2+(x5-
```

Let $c, z_k, w_j \in \mathbb{C}$, $1 \leq k \leq n$, $1 \leq j \leq n-1$. $z_k = x_{2k-1} + x_{2k}i$ $w_j = x_{2n+2j-1} + x_{2n+2j}i$

With constraint, for each j,

$$\sum_k \frac{1}{w_j - z_k} = 0 \Leftrightarrow 0 = \sum_{k=1}^{n} \prod_{\ell=1, \ell \neq k}^{n} (w_j - z_\ell)$$

So that $p(z) = \alpha \prod_{k=1}^{n}(z - z_k)$ and $p'(z) = \beta \prod_{k=1}^{n-1}(w_j - z_\ell)$. Our goal is to seek counterexamples of, or find further numerical evidence for, Borcea's 2-variance conjecture (pg. 4 Khavinson, Pereira, Putinar, Saff, and Shimorin), for each n.

In other words, we seek to minimize

$$(\sigma_2(p))^2 - (h(p,p'))^2.$$

If the above is non-negative for an extremal polynomial, further numerical evidence has been found. If the above is negative for an extremal polynomial, a counterexample to Borcea's 2-variance conjecture has been found.

Along the way, we similarly seek counterexamples of, or further numerical evidence for, Sendov's conjecture.

Where

$$h(p,p') := max_{p(z)=0} min_{p'(w)=0} |z-w|,$$

so that

$$(h(p,p'))^2 := max_{p(z)=0} min_{p'(w)=0} |z-w|^2,$$

and

$$(\sigma_2(p))^2 = min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2 \right).$$

Where

$$c = \frac{1}{n} \sum_{k=1}^{n} z_k.$$

We want to express $\max_k \min_j |z_k - w_j|^2$ as a polynomial, so that the optimization problem

$$min_{c,z_k,w_j} \left(min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2 \right) - \max_k \min_j |z_k - w_j|^2 \right)$$

subject to j contraints

$$0 = \sum_{k=1}^{n} \prod_{\ell=1,\ell\neq k}^{n} (w_j - z_\ell)$$

is one that can be solved using gloptipoly3 or YALMIP.

We recursively express the minimum for fixed k, by using the formula

$$min(a,b) = \frac{a+b-|a-b|}{2}$$

and expressing $|a-b|$ by defining a new variable $y = |a-b|$, so that $y^2 - |a-b|^2 = 0$ and $y \geq 0$.

And we express the maximum as

$$max(c,d) = \frac{c+d+|c-d|}{2}$$

with $z = |c-d|$, $z^2 - |c-d|^2 = 0$ and $z \geq 0$.

4.4 Variables coded into YALMIP

4.5 Degree 2 Example

$n = 2$

$$BasicVariables$$

$$c = x_1 + x_2 i$$

$$z_1 = x_3 + x_4 i$$

$$z_2 = x_5 + x_6 i$$

$$w_1 = x_7 + x_8 i$$

$$Constraint$$

$$(w_1 - z_2) + (w_1 - z_1) = 0$$

$$2x_7 - x_3 - x_5 = 0$$

$$2x_8 - x_4 - x_6 = 0$$

$$VariablesandConstraintsforComputingminimum$$

$$NONE(thereisonlyonew_j)$$

$$VariablesandConstraintsforComputingmaximum$$

$$a = |z_1 - w_1|^2 = (x_3 - x_7)^2 + (x_4 - x_8)^2$$

$$b = |z_2 - w_1|^2 = (x_5 - x_7)^2 + (x_6 - x_8)^2$$

$$x_9 = |a - b|$$

$$x_9^2 = (((x_3 - x_7)^2 + (x_4 - x_8)^2) - ((x_5 - x_7)^2 + (x_6 - x_8)^2))^2$$

$$x_9 \geq 0$$

$$Objective$$

$$|z_1 - c|^2 = |(x_3 - x_1) + (x_4 - x_2)i|^2 = (x_3 - x_1)^2 + (x_4 - x_2)^2$$

$$|z_2 - c|^2 = (x_5 - x_1)^2 + (x_6 - x_2)^2$$

$$min\left(\frac{(x_3 - x_1)^2 + (x_4 - x_2)^2 + (x_5 - x_1)^2 + (x_6 - x_2)^2}{2} - \frac{(x_3 - x_7)^2 + (x_4 - x_8)^2 + (x_5 - x_7)^2 + (x_6 - x_8)^2 + x_9}{2}\right)$$

```
sdpvar x1 x2 x3 x4 x5 x6 x7 x8 x9

F = [2*x7-x3-x5==0,2*x8-x4-x6==0,x9^2-(((x3-x7)^2+(x4-x8)^2)-((x5-x7)^2+(x6-x8)

F = [F, -100 < [x1 x2 x3 x4 x5 x6 x7 x8 x9] < 100];

options=sdpsettings('verbose',1,'solver','bmibnb');

solvesdp(F,(((x3-x1)^2+(x4-x2)^2+(x5-x1)^2+(x6-x2)^2)-((x3-x7)^2+(x4-x8)^2+(x5-
```

4.6 Variables Coded Into YALMIP

In YALMIP, we have $N = 2n^2 + 3n - 1$ real variables x_m. These will be the variables we use in our Semi-Defininte Program optimization procedure. We have a set of feasibility conditions K. And we have an objective $g(\vec{x})$.

We will represent the zeros and critical points as

$$z_k = x_{2k-1} + x_{2k} * i$$

$$w_j = x_{2n+2j-1} + x_{2n+2j} * i$$

In the case of Sendov we add to K, for $2 \leq k \leq n$, the following conditions

$$x_{2k-1}^2 + x_{2k}^2 \leq 1$$

Also, in the case of Sendov, we let remove x_1 and x_2 as semidefinite variables and hard code $z_1 = 1$.

To make sure the w_j are indeed critical points, for each j, we recursively define the constraints via the following pseudo-code.

```
for j from 1 to n-1

for k from 1 to n

for L from 1 to n-1 when L is not j

q=y*

if ell is not

q=y*if ell is not
```

$$|z_k - w_j|^2 = (x_{2k-1} - x_{2n+2j-1})^2 + (x_{2k} - x_{2n+2j})^2$$

$$x_{4n-1} = m_1 = min\{|z_1 - w_1|^2\} = (x_1 - x_{2n+1})^2 + (x_2 - x_{2n+2})^2$$

$$x_{4n} = m_2 = m_{n*(k-1)+j} = min\{|z_1 - w_2|^2, m_1\} \qquad = \frac{1}{2}\big((x_{2*1-1} - x_{2n+2*2-1})^2$$
$$= \frac{1}{2}\left((x_{2*1-1} - x_{2n+2*2-1})^2 + (x_{2*1} - x_{2n+2*2})^2 + x_{4n-1} + x_{4n+1}\right)$$

$$x_{4n+1} \geq 0; x_{4n+1}^2 == (x_{4n-1} - (x_{2k-1} - x_{2n+2j-1})^2 - (x_{2k} - x_{2n+2j})^2)^2$$

$$x_{4n+2*(k-1)*(n-1)+2*(j-2)} = m_\ell = m_{(n-1)*(k-1)+j} = min\{|z_k - w_j|^2, m_{\ell-1}\}$$
$$= \frac{1}{2}\big((x_{2k-1} - x_{2n+2j-1})^2 + (x_{2k} - x_{2n+2j})^2 + m_{\ell-1} + x_{4n+2*(k-1)*(n-1)+2*(}$$
$$= \frac{1}{2}\big((x_{2k-1} - x_{2n+2j-1})^2 + (x_{2k} - x_{2n+2j})^2 + x_{4n+2*(k-1)*(n-1)+2*(j-3)} + x$$

$$x_{4n+2*(k-1)*(n-1)+2*(j-2)+1} \geq 0 x^2_{4n+2*(k-1)*(n-1)+2*(j-2)+1} == (x_{4n+2*(k-1)*(n-1)+2*(j-3)} - (x_{2k-1} - x_{2n}$$

Set $k = 1$ and $j = n - 1$

$$x_{4n+2*(n-3)} = x_{6n-6} = m_{n-1} = min\{|z_1 - w_{n-1}|^2, m_{n-2}\}$$
$$= \frac{1}{2}\left((x_1 - x_{4n-3})^2 + (x_2 - x_{4n-2})^2 + m_{\ell-1} + x_{4n+2*(j-2)+1}\right)$$
$$= \frac{1}{2}\left((x_1 - x_{4n-3})^2 + (x_2 - x_{4n-2})^2 + x_{6n-8} + x_{6n-5}\right)$$

$$x_{4n+2*(n-3)+1} \geq 0 x_{6n-5}^2 == (x_{6n-8} - (x_1 - x_{4n-3})^2 - (x_2 - x_{4n-1})^2)^2$$

Then $x_{6n-6} = \min_j |z_1 - w_j|^2$. This is the maximum so far.

Set $k = 2$ and $j = 1$

$$x_{6n-4} = m_{n-1+1} = m_n = min\{|z_2 - w_1|^2\} = (x_3 - x_{2n+1})^2 + (x_4 - x_{2n+2})^2$$

Set $k = 2$ and $j = 2$ and we see that x_{6n-2} and x_{6n-1} are assigned. So, $x_{4n+2*(k-1)*(n-1)-1}$ is left unassigned. This can be used for the maximum calculation, in the same for loop where the $j = 1$ calculation is done.

When $j = n - 1$, we have the minimum.

$$mm_k = x_{4n+2*(k-1)*(n-1)+2*(n-3)} = \min_j |z_k - w_j|^2$$

$$M_1 = x_{4n+2*(k-1)*(n-1)+2*(n-3)} == x_{4n+2*((K+1)-1)*(n-1)-1}$$

$$M_1 = x_{6n-6} == x_{6n-3}$$

$M > 1$

$$x_{4n+2*k*(n-1)-2} = M_k = \max\{M_{k-1}, mm_k\} = \max\{x_{4n+2*(k-1)*(n-1)-1}, x_{4n+2*(k-1)*(n-1)+2*(n-3)}\}$$
$$= \frac{1}{2}\left(x_{4n+2*(k-1)*(n-1)-2} + x_{4n+2*(k-1)*(n-1)+2*(n-3)} + x_{2*n*(n+1)+k-1}\right)$$

$x_{2*n*(n+1)+k-1} \geq 0 x^2_{2*n*(n+1)+k-1} == (x_{4n+2*(k-1)*(n-1)-2} - x_{4n+2*(k-1)*(n-1)+2*(n-3)})^2$

So that when $k = n$, $x_{2*n*(n+1)-1} = \max_k \min_j |z_k - w_j|^2$.

Now we add our costraint which defines the w_j to be critical points, specifically, for each j

4.7 GLOPTIPOLY

Let $c, z_k, w_j \in \mathbb{C}$, $1 \leq k \leq n$, $1 \leq j \leq n-1$. $c = x_1 + x_2 i$ $z_k = x_{2(k+1)-1} + x_{2(k+1)} i$ $w_j = x_{2(n+1)+2j-1} + x_{2(n+1)+2j} i$

With constraint, for each j,

$$\sum_k \frac{1}{w_j - z_k} = 0 \Leftrightarrow 0 = \sum_{k=1}^{n} \prod_{\ell=1,\ell\neq k}^{n} (w_j - z_\ell)$$

So that $F(z) = a\prod_{k=1}^{n}(z - z_k)$ and $F'(z) = b\prod_{j=1}^{n-1}(z - w_j)$. Our goal is to prove, for each n, Borcea's 2-variance conjecture (pg. 4 Khavinson, Pereira, Putinar, Saff, and Shimorin): $h(F, F') \leq \sigma_2(F)$ or:

$$(\sigma_2(F))^2 - (h(F, F'))^2 \geq 0$$

Where

$$h(F,F') := max_{F(z)=0} min_{F'(w)=0} |z-w|$$

so that

$$(h(F,F'))^2 := max_{F(z)=0} min_{F'(w)=0} |z-w|^2$$

and

$$(\sigma_2(F))^2 = min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2) \right)$$

We want to express $\max_k \min_j |z_k - w_j|^2$ as a polynomial, so that the optimization problem:

$$min_{c,z_k,w_j} \left(min_{c\in\mathbb{C}} \left(\frac{1}{n} \sum_{k=1}^{n} |z_k - c|^2) \right) - \max_k \min_j |z_k - w_j|^2 \right)$$

subject to j contraints:

$$0 = \sum_{k=1}^{n} \prod_{\ell=1,\ell\neq k}^{n} (w_j - z_\ell)$$

is one that can be solved using gloptipoly3.

$$|z_k - w_j|^2 = (x_{2(k+1)-1} - x_{2(n+1)+2j-1})^2 + (x_{2(k+1)} - x_{2(n+1)+2j})^2$$

We 'iteratively' express the minimum for fixed k, by using the formula $min(a,b) = \frac{a+b-|a-b|}{2}$ and expressing $|a-b|$ by defining a new variable $y = |a-b|$, so that $y^2 - |a-b|^2 = 0$ and $y \geq 0$.

$x_{(4n)+1} = \left||z_1 - w_1|^2 - |z_1 - w_2|^2\right| = \left|(x_3 - x_{2(n+1)+1})^2 + (x_4 - x_{2(n+1)+2})^2 - ((x_3 - x_{2(n+1)+3})^2 +\right.$

$$x_{(4n)+1} \geq 0$$

73

$$x_{(4n)+1}^2 - \big((x_3 - x_{2(n+1)+1})^2 + (x_4 - x_{2(n+1)+2})^2 - ((x_3 - x_{2(n+1)+3})^2 + (x_4 - x_{2(n+1)+4})^2)\big)^2 = 0$$

$$x_{(4n)+2} = min(|z_1 - w_1|^2, |z_1 - w_2|^2) = \frac{|z_1 - w_1|^2 + |z_1 - w_2|^2 - x_{(4n)+1}}{2}$$

$$x_{(4n)+3} \geq 0$$

$$x_{(4n)+4} = min(x_{(4n)+2}, |z_1 - w_3|^2)$$

$$x_{(4n-2)+1} = |(x_{2k-1} - x_{2n+2j-1})^2 + (x_{2k} - x_{2n+2j})^2 - \big((x_{2k-1} - x_{2n+2j-1})^2 + (x_{2k} - x_{2n+2j})^2\big)\,|$$

$$x_{=} min(|z_1 - w_1|^2, |z_1 - w_2|^2) =$$

$$min(|z_k - w_j|^2, |z_k - w_{j+1}|^2)$$

A.5 2012 Committee Signatures Of Successful Defense

This was signed subject to a verbal expression of committee members that they would want an updated draft that I promised them.

A.6 Errata to Section A.4

Jón Karl was available to proofread Section A.4 and provided errata, in 2012, which we value and include here. We focus first on providing for our families. So, reader, please pardon the erratic nature of some of this document.

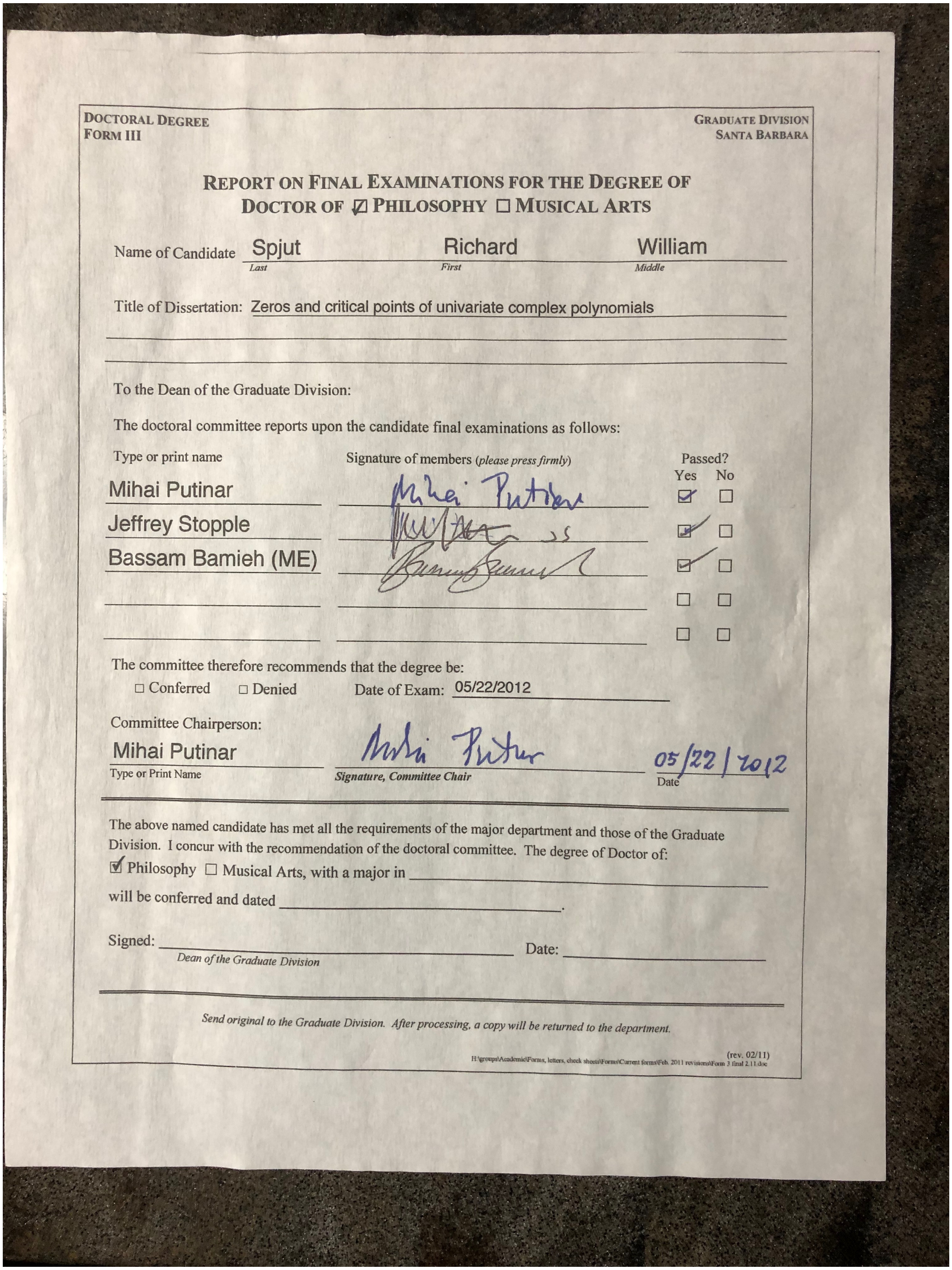
```
Below is a list of few comments that I thought of while reading.

1) In the second to last paragraph in the acknowledgement section you say huge
    gratitude. I find that it is often stronger to understate than so I would
   skip huge.

2) Is it customary to include the curriculum vitae in the dissertation? Also,
   did you intend to write Vit$\ae$ in the title?

3) In remark 2.1.1. I would only use one of the two R \rightarrow \infty or R
   >> 1.

4) I am not sure what information you want to get across when stating examples
    2.1.5-8 but I see you refer to them later. Maybe the examples can be moved
    to the location where they are discussed.
```



DOCTORAL DEGREE
FORM III

GRADUATE DIVISION
SANTA BARBARA

REPORT ON FINAL EXAMINATIONS FOR THE DEGREE OF
DOCTOR OF ☑ PHILOSOPHY ☐ MUSICAL ARTS

Name of Candidate: Spjut (Last) Richard (First) William (Middle)

Title of Dissertation: Zeros and critical points of univariate complex polynomials

To the Dean of the Graduate Division:

The doctoral committee reports upon the candidate final examinations as follows:

Type or print name	Signature of members (*please press firmly*)	Passed? Yes	No
Mihai Putinar		☑	☐
Jeffrey Stopple		☑	☐
Bassam Bamieh (ME)		☑	☐
		☐	☐
		☐	☐

The committee therefore recommends that the degree be:
☐ Conferred ☐ Denied Date of Exam: 05/22/2012

Committee Chairperson:
Mihai Putinar (Type or Print Name) — *Signature, Committee Chair* — 05/22/2012 (Date)

The above named candidate has met all the requirements of the major department and those of the Graduate Division. I concur with the recommendation of the doctoral committee. The degree of Doctor of:
☑ Philosophy ☐ Musical Arts, with a major in ______
will be conferred and dated ______.

Signed: ______ (*Dean of the Graduate Division*) Date: ______

Send original to the Graduate Division. After processing, a copy will be returned to the department.

(rev. 02/11)

Figure A.1: Doctoral Degree Form III from May 22^{nd}, 2012

5) I think it would be clearer to mention in Lemma 2.1.10 that you are referring to the critical point that comes along even though it is obvious from the context.

6) In example 2.2.1 it seems like you are repeating the same thing twice.

7) I do not understand the first paragraph of remark 2.3.1

8) In section 2.4 No Return I do not understand the second to last paragraph on page fifteen and I am not familiar with the notation \int dr/dRdr?

9) There is a bold Code in section 2.5.1. Are you planning on putting code there?

10) I am assuming that you are planning on changing the format of the subsections or 2.5.2 to put them into a more readable format and add more descriptions to the figures and make them more readable.

11) I think I am missing the main part of Chapter 3. How do the acceptable constants of integration relate to the conjecture? Is there a reason why you pick example 3.0.3 to look at in detail?

12) In section 4.1.2. Is it not better to write out fully WLOG then using the abbreviation?

13) In sections 4.1.3-5 the equations do not give me much informations on what is happening. Is there a different way in which you could represent this?

14) Are you going to replace the code in section 4.2.2 by a description of what the code does? I think it would also be good to include more description to what is happening in figures 4.2-4.4.

15) At the beginning of page 61 I would put the definitions of \sigma_2 and h before the paragraph since the paragraph does not have much meaning until you know what \sigma_2 and h are?

16) On page 64-66 you repeat what you have already talked about and the degree 2 example seems to be repeated also.

17) I am not sure what is happening on pages 70 and forward but you are probably still working on them.

This is all shaping up. You know that persistence is the key and remember that a good dissertation is a done dissertation!

Bibliography

[1] Lars V. Ahlfors. *Complex analysis*. Third. An introduction to the theory of analytic functions of one complex variable, International Series in Pure and Applied Mathematics. McGraw-Hill Book Co., New York, 1978, pp. xi+331. ISBN: 0-07-000657-1.

[2] Martin Andersen, Joachim Dahl, and Lieven Vandenberghe. *CVXOPT*. URL: `https://github.com/cvxopt/cvxopt`.

[3] Miguel F. Anjos and Jean B. Lasserre. "Introduction to semidefinite, conic and polynomial optimization". In: *Handbook on semidefinite, conic and polynomial optimization*. Vol. 166. Internat. Ser. Oper. Res. Management Sci. Springer, New York, 2012, pp. 1–22. DOI: `10.1007/978-1-4614-0769-0_1`. URL: `https://doi.org/10.1007/978-1-4614-0769-0_1`.

[4] Joseph Bak and Donald J. Newman. *Complex Analysis*. Third. Undergraduate Texts in Mathematics. Springer, New York, 2010, pp. xii+328. ISBN: 978-1-4419-7287-3. DOI: `10.1007/978-1-4419-7288-0`. URL: `https://doi.org/10.1007/978-1-4419-7288-0`.

[5] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*. Second. Vol. 10. Algorithms and Computation in Mathematics. ISBN 3-540-33098-4. Springer-Verlag, Berlin, 2006, pp. x+662. ISBN: 978-3-540-33098-1.

[6] Grigoriy Blekherman. “Nonnegative polynomials and sums of squares”. In: *J. Amer. Math. Soc.* 25.3 (2012), pp. 617–635. ISSN: 0894-0347. DOI: 10.1090/S0894-0347-2012-00733-4. URL: https://doi.org/10.1090/S0894-0347-2012-00733-4.

[7] Grigoriy Blekherman. “There are significantly more nonnegative polynomials than sums of squares”. In: *Israel J. Math.* 153 (2006), pp. 355–380. ISSN: 0021-2172. DOI: 10.1007/BF02771790. URL: https://doi.org/10.1007/BF02771790.

[8] Grigoriy Blekherman, Pablo Parrilo, and Rehka Thomas. “Semidefinite optimization and convex algebraic geometry”. In: *MOS-SIAM Series on Optimization* 13 (2013).

[9] Jacek Bochnak, Michel Coste, and Marie-Francoise Roy. *Real Algebraic Geometry*. Vol. 36. Springer, 1998, pp. ix+425. ISBN: 3-540-64663-9.

[10] B. D. Bojanov, Q. I. Rahman, and J. Szynal. “On a conjecture of Sendov about the critical points of a polynomial”. In: *Math. Z.* 190.2 (1985), pp. 281–285. ISSN: 0025-5874. DOI: 10.1007/BF01160464. URL: https://doi.org/10.1007/BF01160464.

[11] Iulius Borcea. “On the Sendov conjecture for polynomials with at most six distinct roots”. In: *J. Math. Anal. Appl.* 200.1 (1996), pp. 182–206. ISSN: 0022-247X. DOI: 10.1006/jmaa.1996.0198. URL: https://doi.org/10.1006/jmaa.1996.0198.

[12] Iulius Borcea. “The Sendov conjecture for polynomials with at most seven distinct zeros”. In: *Analysis* 16.2 (1996), pp. 137–159. ISSN: 0174-4747. DOI: 10.1524/anly.1996.16.2.137. URL: https://doi.org/10.1524/anly.1996.16.2.137.

[13] Brian Borchers. “CSDP, a C library for semidefinite programming”. In: vol. 11/12. 1-4. Interior point methods. 1999, pp. 613–623. DOI: 10.1080/10556789908805765. URL: https://doi.org/10.1080/10556789908805765.

[14] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004, pp. xiv+716. ISBN: 0-521-83378-7. DOI: 10.1017/CBO9780511804441. URL: https://doi.org/10.1017/CBO9780511804441.

[15] D. A. Brannan. “On a conjecture of Ilieff”. In: *Proc. Cambridge Philos. Soc.* 64 (1968), pp. 83–85. ISSN: 0008-1981. DOI: 10.1017/s0305004100042596. URL: https://doi.org/10.1017/s0305004100042596.

[16] Johnny E. Brown. “A proof of the Sendov conjecture for polynomials of degree seven”. In: *Complex Variables Theory Appl.* 33.1-4 (1997), pp. 75–95. ISSN: 0278-1077. DOI: 10.1080/17476939708815013. URL: https://doi.org/10.1080/17476939708815013.

[17] Johnny E. Brown and Guangping Xiang. “Proof of the Sendov conjecture for polynomials of degree at most eight”. In: *J. Math. Anal. Appl.* 232.2 (1999), pp. 272–292. ISSN: 0022-247X. DOI: 10.1006/jmaa.1999.6267. URL: https://doi.org/10.1006/jmaa.1999.6267.

[18] M. D. Choi, T. Y. Lam, and B. Reznick. “Sums of squares of real polynomials”. In: *K-theory and algebraic geometry: connections with quadratic forms and division algebras (Santa Barbara, CA, 1992)*. Vol. 58. Proc. Sympos. Pure Math. Amer. Math. Soc., Providence, RI, 1995, pp. 103–126.

[19] Brian Christian and Tom Griffiths. *Algorithms to live by.* The computer science of human decisions. Picador/Henry Holt and Company, New York, 2017, pp. xii+351. ISBN: 978-1-250-11836-3.

[20] Ruel V. Churchill, James W. Brown, and Roger F. Verhey. *Complex variables and applications.* Third. McGraw-Hill Book Co., New York-Düsseldorf-Johannesburg, 1974, pp. x+332.

[21] David A. Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms.* Springer International Publishing, 2015. DOI: 10.1007/978-3-319-16721-3. URL: https://doi.org/10.1007/978-3-319-16721-3.

[22] Raúl E. Curto and Lawrence A. Fialkow. “The truncated complex K-moment problem”. In: *Trans. Amer. Math. Soc.* 352.6 (2000), pp. 2825–2855. ISSN: 0002-9947.

DOI: 10.1090/S0002-9947-00-02472-7. URL: https://doi.org/10.1090/S0002-9947-00-02472-7.

[23] David Damanik. "Singular Continuous Spectrum for Substitution Hamiltonians". PhD thesis. Johann Wolfgang Goethe-Universität Frankfurt am Main, 1998.

[24] Charu Goel, Salma Kuhlmann, and Bruce Reznick. "The analogue of Hilbert's 1888 theorem for even symmetric forms". In: *J. Pure Appl. Algebra* 221.6 (2017), pp. 1438–1448. ISSN: 0022-4049. DOI: 10.1016/j.jpaa.2016.10.003. URL: https://doi.org/10.1016/j.jpaa.2016.10.003.

[25] Robert E. Greene and Steven G. Krantz. *Function Theory of One Complex Variable*. Third. Vol. 40. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2006, pp. x+504. ISBN: 0-8218-3962-4. DOI: 10.1090/gsm/040. URL: https://doi.org/10.1090/gsm/040.

[26] Martin Harrison. "Quadratic Convexity and Sums of Squares". PhD thesis. University of California, Santa Barbara, 2013.

[27] Christoph Helmberg et al. "An interior-point method for semidefinite programming". In: *SIAM J. Optim.* 6.2 (1996), pp. 342–361. ISSN: 1052-6234. DOI: 10.1137/0806020. URL: https://doi.org/10.1137/0806020.

[28] J. William Helton, Scott A. McCullough, and Victor Vinnikov. "NONCOMMUTATIVE CONVEXITY ARISES FROM LINEAR MATRIX INEQUALITIES. J. WILLIAM HELTON, SCOTT A. MCCULLOUGH, AND VICTOR VINNIKOV". In: 2005.

[29] J. William Helton and Mihai Putinar. "Positive polynomials in scalar and matrix variables, the spectral theorem, and optimization". In: *Operator theory, structured matrices, and dilations*. Vol. 7. Theta Ser. Adv. Math. Theta, Bucharest, 2007, pp. 229–306.

[30] Didier Henrion and Jean-Bernard Lasserre. “Detecting global optimality and extracting solutions in GloptiPoly”. In: *Positive polynomials in control.* Vol. 312. Lect. Notes Control Inf. Sci. Springer, Berlin, 2005, pp. 293–310. DOI: 10.1007/10997703_15. URL: https://doi.org/10.1007/10997703_15.

[31] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. “GloptiPoly 3: moments, optimization and semidefinite programming”. In: *Optim. Methods Softw.* 24.4-5 (2009), pp. 761–779. ISSN: 1055-6788. DOI: 10.1080/10556780802699201. URL: https://doi.org/10.1080/10556780802699201.

[32] Anders Nedergaard Jensen. “Computing Gröbner fans and tropical varieties in Gfan”. In: *Software for algebraic geometry.* Vol. 148. IMA Vol. Math. Appl. Springer, New York, 2008, pp. 33–46. DOI: 10.1007/978-0-387-78133-4_3. URL: https://doi.org/10.1007/978-0-387-78133-4_3.

[33] Emmanuel S. Katsoprinakis. “Erratum to: “On the Sendov-Ilyeff conjecture” [Bull. London Math. Soc. **24** (1992), no. 5, 449–455; MR1173941 (93f:30005)]”. In: *Bull. London Math. Soc.* 28.6 (1996). Our local reference was https://arxiv.org/abs/math/0601600, pp. 605–612. ISSN: 0024-6093. DOI: 10.1112/blms/28.6.605. URL: https://doi.org/10.1112/blms/28.6.605.

[34] Dmitry Khavinson et al. “Borcea’s variance conjectures on the critical points of polynomials”. In: *Notions of positivity and the geometry of polynomials.* Trends Math. Birkhäuser/Springer Basel AG, Basel, 2011, pp. 283–309. DOI: 10.1007/978-3-0348-0142-3_16. URL: https://doi.org/10.1007/978-3-0348-0142-3_16.

[35] Steven G. Krantz. *A mathematician’s survival guide.* Graduate school and early career development. American Mathematical Society, Providence, RI, 2003, pp. xvi+222. ISBN: 0-8218-3455-X.

[36] J.-L. Krivine. “Anneaux préordonnés”. In: *J. Analyse Math.* 12 (1964), pp. 307–326. ISSN: 0021-7670. DOI: 10.1007/BF02807438. URL: https://doi.org/10.1007/BF02807438.

[37] Jean B. Lasserre. “A semidefinite programming approach to the generalized problem of moments”. In: *Math. Program.* 112.1, Ser. B (2008), pp. 65–92. ISSN: 0025-5610. DOI: 10.1007/s10107-006-0085-1. URL: https://doi.org/10.1007/s10107-006-0085-1.

[38] Jean B. Lasserre, Monique Laurent, and Philipp Rostalski. “A prolongation-projection algorithm for computing the finite real variety of an ideal”. In: *Theoret. Comput. Sci.* 410.27-29 (2009), pp. 2685–2700. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2009.03.024. URL: https://doi.org/10.1016/j.tcs.2009.03.024.

[39] Jean B. Lasserre and Mihai Putinar. “Positivity and optimization: beyond polynomials”. In: *Handbook on semidefinite, conic and polynomial optimization*. Vol. 166. Internat. Ser. Oper. Res. Management Sci. Springer, New York, 2012, pp. 407–434. DOI: 10.1007/978-1-4614-0769-0_14. URL: https://doi.org/10.1007/978-1-4614-0769-0_14.

[40] Jean Bernard Lasserre. *Moments, positive polynomials and their applications*. Vol. 1. Imperial College Press Optimization Series. Imperial College Press, London, 2010, pp. xxii+361. ISBN: 978-1-84816-445-1; 1-84816-445-9.

[41] Jean Bernard Lasserre, Monique Laurent, and Philipp Rostalski. “Semidefinite characterization and computation of zero-dimensional real radical ideals”. In: *Found. Comput. Math.* 8.5 (2008), pp. 607–647. ISSN: 1615-3375. DOI: 10.1007/s10208-007-9004-y. URL: https://doi.org/10.1007/s10208-007-9004-y.

[42] Monique Laurent. “Sums of squares, moment matrices and optimization over polynomials”. In: *Emerging applications of algebraic geometry*. Vol. 149. IMA Vol. Math.

Appl. Springer, New York, 2009, pp. 157–270. DOI: 10.1007/978-0-387-09686-5_7. URL: https://doi.org/10.1007/978-0-387-09686-5_7.

[43] J. Löfberg. "YALMIP : a toolbox for modeling and optimization in MATLAB". In: *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*. 2004, pp. 284–289.

[44] Johan Löfberg. "A toolbox for modeling and optimization in MATLAB". In: Oct. 2004, pp. 284–289. ISBN: 0-7803-8636-1. DOI: 10.1109/CACSD.2004.1393890.

[45] Johan Löfberg. "Pre- and post-processing sum-of-squares programs in practice". In: *IEEE Trans. Automat. Control* 54.5 (2009), pp. 1007–1011. ISSN: 0018-9286. DOI: 10.1109/TAC.2009.2017144. URL: https://doi.org/10.1109/TAC.2009.2017144.

[46] Nikolai G. Makarov. "Metric Properties of Harmonic Measure". PhD thesis. Steklov Institute of Mathematics, 1986.

[47] Murray Marshall. *Positive polynomials and sums of squares*. Vol. 146. Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI, 2008, pp. xii+187. ISBN: 978-0-8218-4402-1; 0-8218-4402-4. DOI: 10.1090/surv/146. URL: https://doi.org/10.1090/surv/146.

[48] A. Meir and A. Sharma. "On Ilyeff's conjecture". In: *Pacific J. Math.* 31 (1969), pp. 459–467. ISSN: 0030-8730. URL: http://projecteuclid.org/euclid.pjm/1102977881.

[49] M.J. Miller. "Maximal Polynomials and the Ilief-Sendov Conjecture". PhD thesis. University of Rochester, 1984.

[50] H. D. Mittelmann. "An independent benchmarking of SDP and SOCP solvers". In: vol. 95. 2, Ser. B. Computational semidefinite and second order cone programming: the state of the art. 2003, pp. 407–430. DOI: 10.1007/s10107-002-0355-5. URL: https://doi.org/10.1007/s10107-002-0355-5.

[51] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. Vol. 13. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994, pp. x+405. ISBN: 0-89871-319-6. DOI: 10.1137/1.9781611970791. URL: https://doi.org/10.1137/1.9781611970791.

[52] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999, pp. xxii+636. ISBN: 0-387-98793-2. DOI: 10.1007/b98874. URL: https://doi.org/10.1007/b98874.

[53] A. Papachristodoulou et al. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. http://arxiv.org/abs/1310.4716, 2013.

[54] P.A. Parrilo. "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization". PhD thesis. California Institute of Technology, 2000.

[55] Pablo A. Parrilo. "Semidefinite optimization". In: *Semidefinite optimization and convex algebraic geometry*. Vol. 13. MOS-SIAM Ser. Optim. SIAM, Philadelphia, PA, 2013, pp. 3–46.

[56] Pablo A. Parrilo. "Semidefinite programming relaxations for semialgebraic problems". In: vol. 96. 2, Ser. B. Algebraic and geometric methods in discrete optimization. 2003, pp. 293–320. DOI: 10.1007/s10107-003-0387-5. URL: https://doi.org/10.1007/s10107-003-0387-5.

[57] Pablo A. Parrilo and Bernd Sturmfels. "Minimizing polynomial functions". In: *Algorithmic and quantitative real algebraic geometry (Piscataway, NJ, 2001)*. Vol. 60. DIMACS Ser. Discrete Math. Theoret. Comput. Sci. Amer. Math. Soc., Providence, RI, 2003, pp. 83–99.

[58] Dean Phelps and Rene S. Rodriguez. “Some properties of extremal polynomials for the Ilieff conjecture”. In: *Kodai Math. Sem. Rep.* 24 (1972), pp. 172–175. ISSN: 0023-2599. URL: http://projecteuclid.org/euclid.kmj/1138846519.

[59] Daniel Plaumann and Mihai Putinar. “A relative Grace theorem for complex polynomials”. In: *Math. Proc. Cambridge Philos. Soc.* 161.1 (2016), pp. 17–30. ISSN: 0305-0041. DOI: 10.1017/S0305004116000062. URL: https://doi.org/10.1017/S0305004116000062.

[60] Victoria Powers and Thorsten Wörmann. “An algorithm for sums of squares of real polynomials”. In: *J. Pure Appl. Algebra* 127.1 (1998), pp. 99–104. ISSN: 0022-4049. DOI: 10.1016/S0022-4049(97)83827-3. URL: https://doi.org/10.1016/S0022-4049(97)83827-3.

[61] Alexander Prestel and Charles N. Delzell. *Positive polynomials*. Springer Monographs in Mathematics. From Hilbert’s 17th problem to real algebra. Springer-Verlag, Berlin, 2001, pp. viii+267. ISBN: 3-540-41215-8. DOI: 10.1007/978-3-662-04648-7. URL: https://doi.org/10.1007/978-3-662-04648-7.

[62] Q. I. Rahman and G. Schmeisser. *Analytic theory of polynomials*. Vol. 26. London Mathematical Society Monographs. New Series. The Clarendon Press, Oxford University Press, Oxford, 2002, pp. xiv+742. ISBN: 0-19-853493-0.

[63] Michael Reed and Barry Simon. *Methods of modern mathematical physics. I. Functional analysis*. Second. My copy is printed in China by Elsevier (Singapore) Pte Ltd. with Reprint ISBN: 981-4141-65-8 with Copyright dated 2003 and exterior ISBN 7-5062-5931-1. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York, 1980, pp. xv+400. ISBN: 0-12-585050-6.

[64] Bruce Reznick. “Some concrete aspects of Hilbert’s 17th Problem”. In: *Real algebraic geometry and ordered structures (Baton Rouge, LA, 1996)*. Vol. 253. Contemp.

Math. Amer. Math. Soc., Providence, RI, 2000, pp. 251–272. DOI: 10.1090/conm/253/03936. URL: https://doi.org/10.1090/conm/253/03936.

[65] Zalman Rubinstein. "On a problem of Ilyeff". In: *Pacific J. Math.* 26 (1968), pp. 159–161. ISSN: 0030-8730. URL: http://projecteuclid.org/euclid.pjm/1102986035.

[66] E. B. Saff and J. B. Twomey. "A note on the location of critical points of polynomials". In: *Proc. Amer. Math. Soc.* 27 (1971), pp. 303–308. ISSN: 0002-9939. DOI: 10.2307/2036310. URL: https://doi.org/10.2307/2036310.

[67] Richard W. Spjut. *Beamer Slides for 2012 Dissertation Defense*. University of California, Santa Barbara via ProQuest. Supplementary Material. Submitted as one of three Supplementary Materials. May 2012. URL: github.com/spjut/spjut/doctoral_dissertation.

[68] Richard W. Spjut. *Java Project for Generating Scripts for Chapter 4*. 2012,2014 PoormanPublished. 2020 University of California, Santa Barbara via ProQuest. Supplementary Material. One of two Companion Repositories for dissertation. Submitted in Supplementary Materials. May 2012,2014,2020. URL: github.com/spjut/spjut/GeneratesOptimizationScripts.

[69] Richard W. Spjut. *Sage Scripts for Section 3.3.1*. 2018 PoormanPublished. 2020 University of California, Santa Barbara via ProQuest. Supplementary Material. One of two Companion Repositories for dissertation. Submitted in Supplementary Materials. May 2018,2020. URL: github.com/spjut/SageIdealScripts.

[70] Stefan Steidel. "Gröbner bases of symmetric ideals". In: *J. Symbolic Comput.* 54 (2013), pp. 72–86. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2013.01.005. URL: https://doi.org/10.1016/j.jsc.2013.01.005.

[71] Gilbert Stengle. "A nullstellensatz and a positivstellensatz in semialgebraic geometry". In: *Math. Ann.* 207 (1974), pp. 87–97. ISSN: 0025-5831. DOI: 10.1007/BF01362149. URL: https://doi.org/10.1007/BF01362149.

[72] Jan Stochel. "Solving the truncated moment problem solves the full moment problem". In: *Glasg. Math. J.* 43.3 (2001), pp. 335–341. ISSN: 0017-0895. DOI: 10.1017/S0017089501030130. URL: https://doi.org/10.1017/S0017089501030130.

[73] Jos F. Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". In: vol. 11/12. 1-4. Interior point methods. 1999, pp. 625–653. DOI: 10.1080/10556789908805766. URL: https://doi.org/10.1080/10556789908805766.

[74] Bernd Sturmfels. *Gröbner bases and convex polytopes.* Vol. 8. University Lecture Series. American Mathematical Society, Providence, RI, 1996, pp. xii+162. ISBN: 0-8218-0487-1.

[75] R. H. Tütüncü, K. C. Toh, and M. J. Todd. "Solving semidefinite-quadratic-linear programs using SDPT3". In: vol. 95. 2, Ser. B. Computational semidefinite and second order cone programming: the state of the art. 2003, pp. 189–217. DOI: 10.1007/s10107-002-0347-5. URL: https://doi.org/10.1007/s10107-002-0347-5.

[76] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM Rev.* 38.1 (1996), pp. 49–95. ISSN: 0036-1445. DOI: 10.1137/1038003. URL: https://doi.org/10.1137/1038003.

[77] Cynthia Vinzant. "Real Algebraic Geometry in Convex Optimization". PhD thesis. University of California, Berkeley, 2011.

[78] Cynthia Vinzant. "What is ... a spectrahedron?" In: *Notices Amer. Math. Soc.* 61.5 (2014), pp. 492–494. ISSN: 0002-9920. DOI: 10.1090/noti1116. URL: https://doi.org/10.1090/noti1116.

[79] Fei Wang. "Computation of Real Radical Ideals by Semidefinite Programming and Iterative Methods". PhD thesis. The University of Western Ontario, 2016.

[80] Makoto Yamashita, Katsuki Fujisawa, and Masakazu Kojima. "Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0)". In: vol. 18. 4. The Second Japanese-Sino Optimization Meeting, Part II (Kyoto, 2002). 2003, pp. 491–505. DOI: 10.1080/1055678031000118482. URL: https://doi.org/10.1080/1055678031000118482.

[81] Benjamin H. Yandell. *The Honors Class. Hilbert's problems and their solvers.* A K Peters, Ltd., Natick, MA, 2002, pp. x+486. ISBN: 1-56881-141-1.