

DISSERTATION

THE NUMERICAL ALGEBRAIC GEOMETRY APPROACH TO POLYNOMIAL
OPTIMIZATION

Submitted by

Brent R. Davis

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2017

Doctoral Committee:

Advisor: Daniel J. Bates

Co-Advisor: Chris Peterson

Michael Kirby

A.A. Maciejewski

Copyright by Brent R. Davis 2017

All Rights Reserved

ABSTRACT

THE NUMERICAL ALGEBRAIC GEOMETRY APPROACH TO POLYNOMIAL OPTIMIZATION

Numerical algebraic geometry (NAG) consists of a collection of numerical algorithms, based on homotopy continuation, to approximate the solution sets of systems of polynomial equations arising from applications in science and engineering. This research focused on finding global solutions to constrained polynomial optimization problems of moderate size using NAG methods. The benefit of employing a NAG approach to nonlinear optimization problems is that every critical point of the objective function is obtained with probability-one. The NAG approach to global optimization aims to reduce computational complexity during path tracking by exploiting structure that arises from the corresponding polynomial systems. This thesis will consider applications to systems biology and life sciences where polynomials solve problems in model compatibility, model selection, and parameter estimation. Furthermore, these techniques produce mathematical models of large data sets on non-euclidean manifolds such as a disjoint union of Grassmannians. These methods will also play a role in analyzing the performance of existing local methods for solving polynomial optimization problems.

ACKNOWLEDGEMENTS

I would like to thank my PhD advisors, Professors Dan Bates and Chris Peterson for supporting me during the last six years. They have both been extremely generous with the time and funding and I could not imagine achieving what I have without them. Dan has an immense quantity of community outreach. What I have learned from Dan is that volunteering and giving back to the mathematics community is just as important as producing high quality research. He is a true visionary in numerical algebraic geometry and strives at looking toward the future of the area rather than what is going to be in the next publication. Chris has a wealth of knowledge of almost every aspect of mathematics. On any give day, he could show you probably 20 different unique and interesting projects. I really enjoyed working with him in his office and learned so much.

I am thankful for Professors Anton Leykin and Josephine Yu who had organized the 2012 IMA PI Summer Program for Graduate Students. This was one of my first big events as a graduate students and really kick started my interested in conferences in general. One of the highlights of this event was hiking up Stone Mountain, walking through the trees filled with fireflies, and then watching the lazer show projected on the side of the mountain detailing the history of the civil war.

Thanks go to out to the local organizers of MEGA 2013 in Frankfurt, Germany. This was not only my first international conference but also my first time traveling to a foreign country. It was an amazing opportunity and Tara and I especially enjoyed the white asparagus, apfelwein, and visit to Heidelberg during our short time there.

Thanks also go out to the organizers of AG15 in South Korea. This was an extremely well run conference and especially loved exploring the countryside and visiting the Buddhist

temples near Daejeon. One of my highlights was of this trip was visiting Seoul on departure day with Chris Peterson and Hirotachi Abo. On this day we managed traveled to Seoul by train, visit one of the largest Buddhist temples in South Korea, travel around and visit the cities monuments, and just barely have enough time to get to the gate at the airport. I would especially like to thank Abraham Martin Del Campo for all of our interesting conversations we had especially while sipping Soju in South Korea. It has always been a pleasure to have your around and I always look forward to seeing you.

Also thank you to the organizers of the Pingree Park summer school. It was really fun to hear lectures from the leaders of computational algebraic geometry and socialize. One of the highlights of this trip was when Frank Sottile decided on organizing a group to hike up the Mummy Range. In an extremely short amount of time, (a few hours in between lectures), we tried to hike (run), up to the summit all while learning how difficult hiking at such a high elevation in a short amount of time really is. Ironically, Frank Sottile sprained his ankle immediately *after* arriving back to Pingree Park from the hike.

Special thanks to Dr. Andrew Sommese. When Andrew visited CSU on sabbatical he and I spend hours discussing the foundations of numerical algebraic geometry. It was truly an honor to be able to hear directly from one of the creators of the area in a context that I would have never been able to achieve by simply reading books and papers.

I would also like to acknowledge all of the graduate students that were in the basement of Weber during the first couple of years that I was at CSU especially Farrah, Tegan, Tim, Douglas, and Josh. It was really nice to have a good support group in my first few years of graduate school. I would also like to thank everyone in the front office of the Math Department especially Bryan Elder and Sheri Hoefling.

In addition, I would like to acknowledge SIAM for giving me the opportunity to represent CSU at the annual meeting as well as my fellow officers during my two years of service: Tim Marrinan, Farrah Sadre-Mirandi, Tegan Emerson, Melissa Swager, Bahaudin Hasmi, Eric Hanson, Steve Ihde, Sofya Chepushtanova, and Drew Schwickerath.

I would also like to acknowledge all of my collaborators in various forms and degrees: Elizabeth Gross, Heather Harrington, Ken Ho, Michael Kirby, Dan Bates, Chris Peterson, Justin Marks, Matt Niemerg, Eric Hanson, Matt Niemerg, Dani Brake, Tim Hodges, Andrew Sommese, Jon Hauenstein, Joseph Rusinko, Emily Castner, Abraham Martin Del Campo, Jose Rodriguez, Tim Marrinan, and David Ecklund. It was a pleasure to work with all of you.

I also would like to thank my new friends I have gained since moving to Fort Collins especially Kohli, Petrops, and ADB. They provided me with a balance in my life that I was missing in the first couple of years of graduate school and I am grateful for their friendship.

I especially thank my mom, dad, and sister. My mom taught me to never give up even when it felt impossible to move forward. She really motivated me to continue through my third and fourth year of graduate school where I struggled the most. I would also like to thank my dad for his unconditional love and genuine interest in what I have been up to each week. I would also like to thank Sassafrass who was Tara's beloved cat. Sassafrass passed away this year and he will never be forgotten for his gentleness and affection.

The best part about moving to Colorado by far was meeting my girlfriend Tara. Tara has played the largest role in my time here at CSU. Tara had been there supporting me through nearly every major event as a graduate student including my first international conference, part II of my qualifying exam, my preliminary exam, and finally in the last few days before I submit my thesis to my committee by helping me edit and proofread it. She has provided

unconditional love even during my most stressful and pessimistic points along the way. I enjoy seeing all of her successes between receiving her Master's degree, working at CACI, and finally becoming what she always wanted to be: a real forester. All of your success and ability to handle failure has motivated and made me a stronger person. I love you very much.

TABLE OF CONTENTS

Abstract	ii
Acknowledgements	iii
List of Tables	xii
List of Figures	xiv
Chapter 1. Introduction	1
Chapter 2. Mathematical Background	5
2.1. Polynomial Systems	5
2.1.1. Polynomials	6
2.1.2. Algebraic Sets	7
2.1.3. Irreducibility	7
2.1.4. Zariski Topology and Complex Topology	8
2.1.5. Singular Solutions and Multiplicity	8
2.1.6. Probability-One and Genericity	10
2.1.7. Randomization	11
2.1.8. Numerical Approximation and Function Residual	11
2.2. Homotopy Continuation	12
2.2.1. Basic Idea of a Homotopy	13
2.2.2. Homotopies	14
2.2.3. Total-Degree Homotopies	15
2.3. Predictor-Corrector Methods	16
2.3.1. Continuation and Path Tracking	16

2.3.2.	Path Tracking	17
2.3.3.	Predictor-Corrector Methods	18
2.4.	Parameter Homotopy	19
2.5.	Multihomogeneous Homotopy	22
2.5.1.	Multiprojective Space	22
2.5.2.	Multihomogeneous Homotopies	22
2.5.3.	Obtaining a Multihomogeneous Root Count	24
2.6.	Numerical Irreducible Decomposition	25
2.6.1.	Slicing and Degree	25
2.6.2.	Witness Sets and the NID	26
2.6.3.	Computing Witness Sets	27
2.6.4.	Pure-Dimensional Decomposition	28
2.6.5.	Completing the Picture	30
Chapter 3.	Perturbed Regeneration	31
3.1.	Introduction and Motivation	31
3.2.	Mathematical Background	32
3.3.	Regeneration	32
3.4.	Need for Perturbation	36
3.5.	Perturbed Regeneration Algorithm	37
3.6.	Justifying Algorithm 1	38
3.6.1.	Multiplicity	40
3.6.2.	Nonsquare Systems	41
3.7.	Examples and Experimentation	41
3.7.1.	Simple Illustrative Example	41

3.7.2.	cpdm5 System	42
3.7.3.	Fairness of Craps Game	43
3.7.4.	Butcher Problem	45
3.7.5.	Nine-Point Four-Bar Design Problem	47
3.8.	Singular Homotopy Techniques	48
3.8.1.	Regeneration with Deflation	48
3.8.2.	Regenerative Cascade	49
3.8.3.	The Cheater's Homotopy	49
3.9.	Perturbing Positive-Dimensional Components	50
3.9.1.	Failure to Compute Numerical Irreducible Decomposition	50
Chapter 4.	Max-Length Vector Line of Best Fit	52
4.1.	Introduction and Motivation	52
4.2.	Mathematical Background	53
4.2.1.	The Grassmann Manifold and its Representations	53
4.2.2.	Principal Angles between Subspaces	54
4.2.3.	Karcher Mean	55
4.3.	Formulations of the Optimization Problem	55
4.3.1.	Problem Formulation	56
4.3.2.	Geometric Interpretation	58
4.4.	Multivariate Eigenvalue Problem	59
4.4.1.	Solutions to the MLV Line Equations	59
4.4.2.	Iterative Methods	60
4.4.3.	Convergence to Global Solutions	61
4.4.4.	Degenerate Cases	61

4.4.5.	Measure of Correlation	62
4.5.	A Multivariate Eigenvalue Homotopy	63
4.5.1.	MEV Homotopy	64
4.5.2.	Comparison of Root Counts	66
4.6.	Examples applying the MLV Line	67
4.6.1.	Small Example	67
4.6.2.	Large Example	67
4.6.3.	Iterative Method	68
4.6.4.	Application to Image Data	69
Chapter 5.	Model Selection	73
5.1.	Introduction and Motivation	73
5.2.	Problem Statement	74
5.2.1.	Model Validation	75
5.2.2.	Model Selection	79
5.2.3.	Parameter Estimation	79
5.3.	Geometry	80
5.3.1.	NAG Techniques Used	82
5.4.	Algorithms	82
5.4.1.	Algorithm: Model Validation	84
5.4.2.	Nonnegativity Considerations	85
5.4.3.	Algorithm: Model Selection	87
5.4.4.	Algorithm: Parameter Estimation	88
5.4.5.	Illustrative Example	88
5.5.	Results and Experiments	90

5.5.1. Cell Death Activation	90
5.5.2. HIV Progression	98
5.5.3. Multisite Phosphorylation.....	101
Chapter 6. Conclusion	114
Bibliography	116

LIST OF TABLES

3.1	Basic properties of the cpdm5 problem.....	43
3.2	Timings for the cpdm5 problem.....	43
3.3	Paths tracked for the cpdm5 problem.....	43
3.4	Timings for the unfair dice problem.....	45
3.5	Paths tracked for the unfair dice problem.....	45
3.6	Irreducible components of the Butcher problem.....	46
3.7	Timings for the Butcher problem.....	46
3.8	Timings for the nine point problem.....	47
4.1	Root counts of MLV solution methods.....	67
5.1	Timings for clustering model.....	96
5.2	Reactions for HIV model.....	99
5.3	Variables for MAP Kinase models.....	102
5.4	Parameters for MAP Kinase models.....	103
5.5	Reaction velocities for MAP Kinase models.....	103
5.6	Equations for MAP Kinase models.....	104
5.7	Parameter values for MAP Kinase models.....	104
5.8	EGF loading data.....	106
5.9	Path counts for MAP Kinase models.....	110
5.10	Timings for MAP Kinase models.....	110
5.11	Test statistic for distributive model.....	111

5.12 Test statistic for processive model..... 112

LIST OF FIGURES

4.1	Appending the unknown subject	70
4.2	Changing basis of subspaces	70
4.3	MLV line on data	71
5.1	NAG framework for model selection	83
5.2	Illustrative example for model validation	89
5.3	Test statistics for MAP Kinase models versus EGF loading	113

CHAPTER 1

INTRODUCTION

Numerical algebraic geometry (NAG) consists of a collection of numerical algorithms based on homotopy continuation to approximate solution sets to systems of polynomial equations arising from applications in science and engineering [9, 69]. The study of NAG was initially interested in solving a host of difficult problems in robot kinematics. Over the last twenty years, NAG has revolutionize how engineers approach robot kinematics problems [69, 68, 70]. Most notably was the complete solution to the nine-point problem for four-bar linkages, a fundamental problem in robot kinematics [78]. The complete solution was solved in 1992 using NAG methods nearly 70 years after its initial problem statement.

Beyond robot kinematics there has been a push to expand NAG into other applications including (but not excluding): optimal control [66], nonlinear ordinary differential equations [2], vibrations of thin plates [53], guage and string theory [56], necrotic tumor models [35], and multicomponent mixture models [67].

I have focused my research on finding global solutions to constrained polynomial optimization problems of moderate size using NAG methods. The motivation for applying NAG methods is that the global solution is always obtained using probability-one arguments from NAG. There is no guarantee that traditional local methods will perform well because a given optimization problem may contain a large number of critical points.

For example, an iterative gradient descent method may get trapped in a region where a local nonglobal critical point is obtained instead of a global critical point. Another example arises from the area of binary integer programming where the optimization problems that

are proven to be NP-hard such as those appearing in Karp's famous 21 NP-complete problems [45]. Polynomial optimization problems are difficult to solve in general but techniques can be applied to solve some medium-sized problems arising from applications.

The benefit of employing a NAG approach to nonlinear optimization problems is that every isolated critical point of the objective function is obtained with probability-one. A global solution is then obtained simply by evaluating the objective function at every critical point and selecting the one with the minimum objective value. There are also techniques for polynomial optimization using semidefinite programming which aim to solve a related optimization problem using convex relaxations (see [15] and reference within), there is no guarantee that these methods will find the global solution. However, these methods do provide a certification if a global solution is found using a duality principle [15]. Recent trends in this area have been aimed at exploiting the sparsity structure of the objective function and constraints [15]. This approach helps increase the applicability to a variety of new and challenging problems [15].

Just as semidefinite programming aims to minimize computational complexity, the NAG approach to global optimization also aims to reduce computational complexity during path tracking by exploiting structure that arises from the corresponding polynomial systems [9, 69]. For example, a common structure arises from solving parameterized optimization problems where a large set of related optimization problems must be solved. Here, a parameter homotopy approach from NAG is most appropriate in solving the corresponding polynomial equations and drastically reduces the average cost to solve systems at each parameter value [16].

In addition, if the optimization problem is constrained one can use a Lagrange multiplier method whose equations are linear in the Lagrange multiplier variables. In this case a multi-homogeneous homotopy from NAG significantly reduces path tracking during the homotopy continuation routine of NAG [9, 69].

Polynomials arising from optimization problems that are suited for NAG methods arise in the fields of systems biology and life sciences where polynomials solve problems in model compatibility, model selection, and parameter estimation [5]. For example, given a prescribed set of experimental output data one can select from a collection of mathematical models that best describe multisite phosphorylation mechanisms [5]. This information can then be used to explain how these biological mechanisms behave differently *in vivo* versus *in vitro* [4].

Furthermore, optimization problems where NAG may be applied arise in geometric data analysis problems [6]. Solutions to polynomials construct the minimum arguments of a prescribed optimization problem [6]. One can then produce mathematical models of large data sets on non-euclidean manifolds such as a disjoint union of Grassmannians. Modeling data on a Grassmannian has the benefit of preserving orthogonal invariance than if relevant in data sets where illumination plays a large role in data clustering subroutines. The maximum length vector line of best fit can be defined in such a way as to extract common features of a data set hidden from the naked eye [6]. As will be shown, the polynomial system that arises is the so-called multivariate eigenvalue problem; a generalization of the traditional eigenvalue problem [22].

In addition, NAG plays a role in analyzing the performance of existing local methods for solving polynomial optimization problems where they may have issues of obtaining the global solution [6]. By comparing solutions obtained using NAG methods to local methods, one can analyze what proportion of the time the local methods found the global solutions [6].

Chapter 2 contains a survey the major mathematical background from the field of NAG as it relates to this dissertation. Much of NAG is documented in the two textbooks: *Numerically solving polynomial systems with Bertini* [9] and *The numerical solution of systems of polynomials arising in engineering and science* [69]. This chapter lays the foundation for the chapters that follow which use concepts such as: genericity, randomization, good homotopies, intersection geometry, and the numerical irreducible decomposition.

Chapter 3 contains a novel mathematical technique called *perturbed regeneration*. Perturbed regeneration is a method to compute all isolated solutions to polynomial systems of equations including singular solutions. This method works especially well in solving systems of equations arising from optimization problems especially when a large set of parameter values are considered.

Chapter 4 contains a new mathematical model for data sets called the *max-length vector line of best fit* (MLV line). The MLV line aims to described a collection of data sets arising as points on a Grassmannian manifold.

Finally, chapter 5 discusses a new paradigm for *model selection*. In this chapter, we solve three fundamental problems in science: model validation, model selection, and parameter estimation. The NAG approach to model selection is built on geometric principles. The techniques not only use NAG to find every critical point of corresponding optimization problems but also uses a fundamental structure in NAG, the numerical irreducible decomposition, to determine intersections of algebraic sets that represent compatibility of model with prescribed data.

CHAPTER 2

MATHEMATICAL BACKGROUND

The aim of this chapter is to discuss the numerical algebraic geometry (NAG) background necessary for Chapter 3–5¹. A discussion of polynomial systems in general is found in §2.1. The foundations of how to interpret and explain solutions to polynomial systems are laid out. In §2.2 homotopy continuation is discussed. Homotopy is the main mechanism that all NAG algorithms are built on. In §2.3 predictor-corrector methods bridge the gap between theory and computation. Predictor-corrector methods are the means to realize homotopy continuation using numerical approximation and solve problems from applications. Then in §§2.4–2.5 more specific homotopy routines are discussed that arise from applications and optimization; the parameter and multihomogeneous homotopies. Finally, the numerical irreducible decomposition is discussed in §2.6 where NAG is employed to handle positive-dimensional solutions such as curves and surfaces.

2.1. POLYNOMIAL SYSTEMS

This section discusses the relevant aspects of polynomials that will be employed throughout the remaining chapters. §2.1.1 discusses a basic introduction to polynomials. It is then followed by §§2.1.2–2.1.3 on algebraic sets and irreducibility. The Zariski topology is discussed in §2.1.4 which lay the foundation for genericity arguments and randomization in §§2.1.6–2.1.7. Singular solutions and multiplicity are described in §2.1.5. Finally, aspects of numerical approximation are discussed in §2.1.8.

¹Nearly the entirety of material from this chapter is taken from two very useful sources: *Numerically solving polynomial systems with Bertini* [9] and *The numerical solution of systems of polynomials arising in engineering and science* [69]. These are the current textbooks on numerical algebraic geometry to date. Citations outside these references are done normally within the text.

2.1.1. POLYNOMIALS. We are interested in computing the solution set of a polynomial system of equations:

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_n(z_1, \dots, z_N) \end{bmatrix} = \mathbf{0}$$

where each $f_i(\mathbf{z})$ is a multivariable polynomial function with complex coefficients in the variables $\mathbf{z} = (z_1, \dots, z_N)$. Algebraically, $\mathbf{f}_i(\mathbf{z}) \in \mathbb{C}[z_1, \dots, z_N]$, the polynomial ring in the indeterminants \mathbf{z} over the field \mathbb{C} . Here, N could be different from n . When $N = n$, we say the system is *square*. When $N > n$, there are more variables than equations and we say the system is *underdetermined*. When $N < n$, the system is said to be *overdetermined*. Square systems are important to NAG and often times we can “square up” the system using randomization.

For example, a degree two polynomial in variables x, y has the form:

$$\square x^2 + \square xy + \square y^2 + \square x + \square y + \square$$

where ‘ \square ’ is shorthand notation and designates a complex number that may be different on each of the six terms.

EXAMPLE 2.1.1. Consider the polynomial system defined by:

$$\mathbf{f}(x, y) = \begin{bmatrix} x(x - 1) \\ x(y - 1) \end{bmatrix}.$$

The solutions of $\mathbf{f}(x, y)$ consist of the y -axis (where $x = 0$) together with the isolated point $(x, y) = (1, 1)$. As this simple example demonstrates, polynomial systems exhibit solutions that are more complicated than solutions to systems of linear equations where there is either exactly one solution, no solutions, or infinitely many solutions expressed as a linear subspace of Euclidean space.

2.1.2. ALGEBRAIC SETS. An *affine complex algebraic set*:

$$\mathcal{V}(f_1, \dots, f_n) = \{\mathbf{z} \in \mathbb{C}^N \mid f_1(\mathbf{z}) = \dots = f_n(\mathbf{z}) = 0\}$$

is a locus of solutions on \mathbb{C}^N of a system of polynomials with complex coefficients. By the fundamental theorem of algebra, any degree d univariate polynomial has exactly d roots, counting multiplicities [24]. In the multivariable case, we have a similar result. Hilbert's Nullstellensatz states that if $\mathcal{V}(f_1, \dots, f_n) \subset \mathbb{C}^N$ is empty, there exists polynomials g_1, \dots, g_n such that:

$$f_1 g_1 + \dots + f_n g_n = 1.$$

A *smooth point* $\mathbf{p}^* = (p_1^*, \dots, p_N^*)$ of $X = \mathcal{V}(f_1, \dots, f_n)$ is a point $\mathbf{p}^* \in X$ with a nonempty neighborhood $U \subset X$ such that for some mapping $\Phi(z_1, \dots, z_N)$, $\Phi|_U$ maps U bijectively to a neighborhood of the origin in \mathbb{C}^k for some k . The set of smooth points is often denoted X_{reg} and the *complex dimension* of X at \mathbf{p}^* is k .

2.1.3. IRREDUCIBILITY. A single polynomial $f(\mathbf{z})$ is said to be *irreducible* if it cannot be factored as $f = gh$ where g, h are nonconstant polynomials. A polynomial $f(\mathbf{z})$ is irreducible if and only if $\mathcal{V}(f)_{\text{reg}}$ is connected as a complex manifold. Over the complex numbers, for any algebraic set X , the closure of any connected component of X_{reg} is also an algebraic set. The component may therefore be cut out by a collection of polynomials.

Every polynomial f may be factored as $f = p_1^{m_1} \dots p_\ell^{m_\ell}$ where $m_j \in \mathbb{N}_{>0}$ and each p_j is an irreducible polynomial. Then one has a decomposition:

$$\mathcal{V}(f) = \bigcup_{j=1}^{\ell} \mathcal{V}(p_j).$$

$\mathcal{V}(f)$ decomposes as a union of closures of the complex connected components of $\mathcal{V}(f)$. An affine complex algebraic set $X = \mathcal{V}(f_1, \dots, f_n) \subset \mathbb{C}^N$ is *irreducible* if X_{reg} is connected.

X_{reg} is the complement of an affine complex algebraic set, therefore X_{reg} is dense in X . For $\mathbf{p}^* \in X$, the *dimension of X at \mathbf{p}^** is the maximum dimension of the irreducible components that pass through \mathbf{p}^* , denoted $\dim_{\mathbf{p}^*} X$. The *dimension of X* is:

$$\dim X = \max\{\dim_{\mathbf{p}^*} X \mid \mathbf{p}^* \in X\}.$$

An affine complex algebraic set X is called *pure-dimensional* if $\dim X = \dim_{\mathbf{p}^*} X$ for all $\mathbf{p}^* \in X$. For any non-constant polynomial f , the complex algebraic set $\mathcal{V}(f)$ has dimension $N - 1$ at every point of $\mathcal{V}(f)$ and we call $\mathcal{V}(f)$ a *hypersurface*. We discuss the irreducible decomposition in §2.6.

2.1.4. ZARISKI TOPOLOGY AND COMPLEX TOPOLOGY. Complex algebraic sets $X \subset \mathbb{C}^N$ have a *complex topology* where open sets are unions of the intersections of X with open balls centered at points of \mathbb{C}^N . We may also place a *Zariski topology* on X with open set defined as the intersection of X with complements of complex algebraic subsets of \mathbb{C}^N . An open set in the complex topology need not be an open set in the Zariski topology.

If U is a Zariski open subset of X the closure of X in both the complex and Zariski topologies are equal. Also if X is irreducible then every nonempty Zariski open subset $U \subset X$ is path connected. U is the complement in X of a complex codimension one subset Z of X where Z has Lebesgue measure zero. Often Z referred to as a “thin” subset.

2.1.5. SINGULAR SOLUTIONS AND MULTIPLICITY. As illustrated in example 2.1.1 polynomials can exhibit solution sets that are zero-dimensional (points) or positive-dimensional (curves, surfaces, ect.). A solution $\mathbf{z}^* \in \mathbb{C}^N$ to a polynomial system $\mathbf{f}(\mathbf{z})$ is *isolated* if there exists a $r > 0$ such that \mathbf{z}^* is the only solution in an open ball of radius r centered at \mathbf{z}^* . Isolated solutions may be either *nonsingular* or *singular* and depend on the defining polynomials.

Consider a general univariate polynomial $f(z)$ with solution z^* . We say the *multiplicity* of the solution z^* of $f(z)$ is k if $f^{(j)}(z^*) = 0$ for $0 \leq j < k$ and $f^{(k)}(z^*) \neq 0$. Singular solutions to univariate polynomials have multiplicity at least two.

In the case for multivariate systems of polynomials multiplicity is more difficult to define but there is still a notion of singular and nonsingular. Consider a multivariate polynomial $\mathbf{f}(\mathbf{z})$ in N variables and n equations. We say a solution \mathbf{z}^* is *nonsingular* if \mathbf{z}^* is a solution to $\mathbf{f}(\mathbf{z})$ and the Jacobian matrix of $\mathbf{f}(\mathbf{z})$ at \mathbf{z}^* denoted $J\mathbf{f}(\mathbf{z}^*)$ is full rank. That is, $\text{rank } J\mathbf{f}(\mathbf{z}^*) = \min(N, n)$. Otherwise \mathbf{z}^* is *singular* and the Jacobian matrix drops rank. One useful fact is that if \mathbf{z}^* is a solution to $\mathbf{f}(\mathbf{z})$ and on a positive-dimensional component then \mathbf{z}^* is always a singular solution but could be a smooth point of the corresponding complex manifold.

Numerical methods to compute approximate solutions to polynomial systems behave poorly near singular solutions. The Newton-Raphson method is an iterative method to compute numerical approximations to solutions of nonlinear systems of equations. Given a starting guess \mathbf{z}_i for a solution of $\mathbf{f}(\mathbf{z})$, Newton's method produces an updated guess:

$$\mathbf{z}_{i+1} = \mathbf{z}_i - [J\mathbf{f}(\mathbf{z}_i)]^{-1}\mathbf{f}(\mathbf{z}_i)$$

with Newton residual $\|\mathbf{z}_{i+1} - \mathbf{z}_i\|$. Consider a polynomial system $\mathbf{f}(\mathbf{z})$ with nonsingular solution \mathbf{z}^* and an initial guess \mathbf{z}_0 sufficiently close to \mathbf{z}^* . Newton's method will converge *quadratically* to \mathbf{z}^* . That is:

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{z}^{k+1} - \mathbf{z}^*\|}{\|\mathbf{z}^k - \mathbf{z}^*\|^2} < \infty.$$

Quadratic convergences of Newton's method does not necessarily hold for singular solutions. There does not exist any neighborhood of \mathbf{z}^* such that Newton's method with starting guess \mathbf{z}_0 will converge quadratically to the solution \mathbf{z}^* . In some cases while implementing Newton's

method iterates will diverge away from the solution \mathbf{z}^* such as the Griewank and Osborn system will demonstrate in the following example. Quadratic convergence plays a large role in the definition for approximate solutions to polynomials.

EXAMPLE 2.1.2. Consider the Griewank and Osborn system [31]:

$$\mathbf{g}(x, y) = \begin{bmatrix} \frac{29}{16}x^3 - 2xy \\ y - x^2 \end{bmatrix}.$$

The solution set of $\mathbf{g}(x, y)$ has only one singular solution $(0, 0)$. Even if a starting point $(x, y) = (x_0, y_0)$ is arbitrarily close to but not equal to $(0, 0)$, it have been proven that Newton's Method will diverge away from $(0, 0)$.

2.1.6. PROBABILITY-ONE AND GENERICITY. A statement $\mathcal{P}(\mathbf{z})$ parameterized by \mathbf{z} , where \mathbf{z} belongs to a nonempty irreducible complex algebraic set X , is said to be *generically true* or *true with probability one* if $\mathcal{P}(\mathbf{z})$ is true for all \mathbf{z} in a nonempty Zariski open subset of X . If X is irreducible then a nonempty Zariski open subset $U \subset X$, which is the complement in X of a complex codimension one subset, is connected. The closure of U is X and $X \setminus U$ is a measure one set. This means that a property $\mathcal{P}(\mathbf{z})$ holds for essentially all of X . An example of genericity in action is Bertini's Theorem.

THEOREM 2.1.1. (Bertini's Theorem) Suppose $f_1(\mathbf{z}), \dots, f_n(\mathbf{z})$ are polynomials defined on a nonempty Zariski open subset U of an irreducible affine algebraic set $X \subset \mathbb{C}^N$ so that given any point $\mathbf{z} \in U$, there is at least one i such that $f_i(\mathbf{z}) \neq 0$. Then there exists a nonempty Zariski open subset $V \subset \mathbb{C}^n$ such that for $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in V \subset \mathbb{C}^n$, $\mathbf{f}_{\boldsymbol{\lambda}}(\mathbf{z}) = \lambda_1 f_1(\mathbf{z}) + \dots + \lambda_n f_n(\mathbf{z})$ has the property that $\mathcal{V}(\mathbf{f}_{\boldsymbol{\lambda}}(\mathbf{z}))$ is empty or pure dimension $N - 1$ and $\mathcal{V}\left(\mathbf{f}_{\boldsymbol{\lambda}}(\mathbf{z}), \frac{\partial \mathbf{f}_{\boldsymbol{\lambda}}(\mathbf{z})}{\partial z_1}, \dots, \frac{\partial \mathbf{f}_{\boldsymbol{\lambda}}(\mathbf{z})}{\partial z_N}\right) = \emptyset$.

An extension of Bertini's Theorem tell us that there is a nonempty Zariski open subset U of $k \times n$ matrices $\mathbb{C}^{k \times n}$ such that for any $A \in U$ then $\mathcal{V}(\mathbf{A}\mathbf{f})$ is empty or pure dimension $N - k$.

2.1.7. RANDOMIZATION. In §2.2 techniques of homotopy continuation are described to solve polynomial systems of equations. However, the number of equations and variables must be the same to employ the methods. Given a collection of polynomials f_1, \dots, f_n , a *random linear combination* has the form:

$$\square f_1 + \dots + \square f_n$$

where the \square 's are say, independently and identically distributed (i.i.d.) draws of complex numbers whos moduli are sampled from a Gaussian distribution with mean $\mu = 1$ and small variance σ . The arguments of these complex numbers are sampled uniformly from the interval $[0, 2\pi]$. Often times, we will say the \square 's are *chosen randomly*, but keep this notion of randomness in mind.

To randomize an overdetermined system ($N < n$), we replace the polynomials f_1, \dots, f_n with N random linear combinations of the polynomials. That is, for a random matrix $\mathbf{A} \in \mathbb{C}^{N \times n}$ we replace the system $\mathbf{f}^T = (f_1, \dots, f_n)^T$ with the system $\mathcal{R}(\mathbf{f}) = \mathbf{A}\mathbf{f}$ where we say that $\mathcal{R}(\mathbf{f})$ is a *randomization of \mathbf{f}* . Randomizations of \mathbf{f} are often referred to as “squaring up the system”. If \mathbf{z}^* is a solution to $\mathbf{f}(\mathbf{z})$, then it is also a solution to $\mathcal{R}(\mathbf{f})(\mathbf{z})$ according to and application of theorem 2.1.1.

2.1.8. NUMERICAL APPROXIMATION AND FUNCTION RESIDUAL. Given a polynomial system $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ suppose \mathbf{z}^* is a solution to $\mathbf{f}(\mathbf{z})$. We want to compute a complex vector $\hat{\mathbf{z}}$ so that:

$$\|\mathbf{z}^* - \hat{\mathbf{z}}\| < \epsilon$$

for some sufficiently small $\epsilon > 0$ using the Euclidean 2-norm, for example. In this case, we say that $\hat{\mathbf{z}}$ is an *approximate solution* to $\mathbf{f}(\mathbf{z})$. Note that an alternate definition of approximate solution is to use quadratic convergence of Newton's method as a definition discussed briefly in §2.1.5. We must be careful in quantifying approximate solution since even though the distance may be small, the function residual may be quite large.

LEMMA 2.1.1. [69] Given a univariate polynomial $p(z) = z^d + \square z^{d-1} + \dots + \square$ and $\epsilon > 0$, the area of the set of $z \in \mathbb{C}$ such that $|p(z)| \leq \epsilon$ is at most $d\pi\epsilon^{2/d}$.

Although this estimate is not sharp we may interpret lemma 2.1.1 as stating that as the degree d increases or the tolerance ϵ on the function residual is tightened the regions where the function residual is small shrinks.

2.2. HOMOTOPY CONTINUATION

Homotopy continuation is the fundamental computation in NAG. Consider the problem of finding the finite solutions of a square system:

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = \mathbf{0}$$

for $\mathbf{z} \in \mathbb{C}^N$. Square systems are relevant to all aspects of NAG ranging from theory to applications. For example, if we can find solutions to square systems then overdetermined systems are easy to handle due to randomization.

In §2.2.1 we will discuss the high-level details of how a homotopy is used to find solutions to polynomial systems. In §2.2.2 we discuss homotopies in greater detail and define *good homotopies*, and finally in §2.2.3 we discuss the total-degree homotopy which is a canonical good homotopy that is always ensured to find isolated solutions with probability-one.

2.2.1. BASIC IDEA OF A HOMOTOPY. In order to “solve” a polynomial system of equations $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ first solve a similiar and related system $\mathbf{g}(\mathbf{z}) = \mathbf{0}$ that is “easy” to solve and then “deform” the solutions of \mathbf{g} to \mathbf{f} . First form a one-real parameter family of polynomials called a homotopy. Homotopies are the mechanism to allow us to deform solutions of \mathbf{g} to \mathbf{f} . There are three steps used to solve a polynomial system $\mathbf{f}(\mathbf{z})$:

- (1) (**Construct and solve a start system $\mathbf{g}(\mathbf{z})$**) First construct a start system $\mathbf{g}(\mathbf{z})$ whose properties are “similiar” enough to $\mathbf{f}(\mathbf{z})$. The solutions of $\mathbf{g}(\mathbf{z})$ are called *start points*.
- (2) (**Construct a homotopy between $\mathbf{f}(\mathbf{z})$ and $\mathbf{g}(\mathbf{z})$**) Define a homotopy function. One such example is the straight-line homotopy $\mathbf{H}(\mathbf{z}, t) = t\mathbf{g}(\mathbf{z}) + (1 - t)\mathbf{f}(\mathbf{z})$ where t is some parameter. Note that $\mathbf{H}(\mathbf{z}, 1) = \mathbf{g}(\mathbf{z})$ and $\mathbf{H}(\mathbf{z}, 0) = \mathbf{f}(\mathbf{z})$ so that the homotopy recovers the “start” and “target” systems at these parameters. For correctly chosen $\mathbf{g}(\mathbf{z})$ the solutions of $\mathbf{H}(\mathbf{z}, t) = \mathbf{0}$ as t varies in the interval $(0, 1]$ will be smooth and vary continuously with probability-one. These define smooth strictly increasing paths from the start points of $\mathbf{g}(\mathbf{z})$ and the target solutions of $\mathbf{f}(\mathbf{z})$.
- (3) (**Track paths from $t = 1$ to $t = 0$**) Use predictor-corrector methods described in §2.3 to follow paths from $t = 1$ to $t = 0$. Path tracking using predictor-corrector methods is usually done on a subset of the interval $(0, 1]$, and then the so-called endgames are employed to track paths as $t \rightarrow 0$. Often times, the number of solutions of $\mathbf{f}(\mathbf{z})$ is far fewer that the number of solutions of $\mathbf{g}(\mathbf{z})$ in which case we need to address divergent paths.

2.2.2. HOMOTOPIES. A *homotopy* is a smooth function:

$$\mathbf{H}(\mathbf{z}, t) : \mathbb{C}^N \times [0, 1] \rightarrow \mathbb{C}^N.$$

For our purposes, \mathbf{H} is obtained via a sequence of compositions first from a family of systems $\mathcal{H}(\mathbf{z}; \mathbf{s})$ where $\mathbf{s} \in U \subset \mathbb{C}^M$ and $\mathbf{s} = \mathbf{q}(t)$ for a real parameter t . More precisely we have:

- (1) A mapping $\mathcal{H}(\mathbf{z}; \mathbf{s}) : \mathbb{C}^N \times U \rightarrow \mathbb{C}^N$, where U is an open subset of \mathbb{C}^M and the coordinates are polynomial functions in the variables \mathbf{z} and complex analytic in the parameters \mathbf{s} .
- (2) A differentiable mapping $\mathbf{q} : [0, 1] \rightarrow U$ which is the path parameterized by t going from $\mathbf{s}_1 = \mathbf{q}(1)$ to $\mathbf{s}_0 = \mathbf{q}(0)$ in the parameter space U .

We then construct \mathbf{H} by taking the composition $\mathbf{H}(\mathbf{z}, t) := \mathcal{H}(\mathbf{z}; \mathbf{q}(t)) : \mathbb{C}^N \times [0, 1] \rightarrow \mathbb{C}^N$.

Tracking is the process of approximating paths with the goal of obtaining the solutions of $\mathbf{H}(\mathbf{z}, 0)$. The details of predictor-corrector methods are discussed in §2.3. A *good homotopy* $\mathbf{H}(\mathbf{z}, t)$ for the system:

$$\mathbf{f}(\mathbf{z}) = \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} = \mathbf{0}$$

together with a set of D solutions S_1 of $\mathbf{g}(\mathbf{z}) = \mathbf{H}(\mathbf{z}, 0)$ is a homotopy of infinitely differentiable functions:

$$\mathbf{H}(\mathbf{z}, t) = \begin{bmatrix} H_1(z_1, \dots, z_N) \\ \vdots \\ H_N(z_1, \dots, z_N) \end{bmatrix}$$

such that:

- (1) For any choice of $t \in [0, 1]$, $\mathbf{H}(\mathbf{z}, t)$ is a polynomial system.
- (2) For any of the D start points $\mathbf{w}_j \in S_1$, for $1 \leq j \leq D$, there is a smooth mapping $\mathbf{p}_j(t) : (0, 1] \rightarrow \mathbb{C}^N$ so that $\mathbf{p}_j(1) = \mathbf{w}_j$ and has the property that,

(3) \mathbf{p}_j is smooth on $(0,1]$ and for each $t^* \in (0,1]$, there does not exist distinct integers $1 \leq j, k \leq D$ and $t^* \in (0,1]$ so that $\mathbf{p}_j(t^*) = \mathbf{p}_k(t^*)$ and in addition points $\mathbf{p}_j(t^*)$ are smooth isolated solutions of $\mathbf{H}(\mathbf{z}, t^*)$.

(4) Choose D startpoints of S_1 of $\mathbf{g}(\mathbf{z}) = \mathbf{H}(\mathbf{z}, 1) = \mathbf{0}$ so that the set:

$$S_0 = \{\mathbf{z} \in \mathbb{C}^N \mid \|\mathbf{z}\|_2 < \infty \text{ and } \mathbf{z} = \lim_{t \rightarrow 0} \mathbf{p}_j(t)\}$$

contains every isolated solution of $\mathbf{f}(\mathbf{z})$.

Essentially this technical definition states that a good homotopy $\mathbf{H}(\mathbf{z}, t)$ for a system $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ is one that will find all isolated solutions of $\mathbf{f}(\mathbf{z})$ using paths that “vary smoothly” and don’t interfere with other paths on the interval $(0,1]$ and produce the solutions of $\mathbf{f}(\mathbf{z})$ as $t \rightarrow 0$. One useful property is that the paths satisfy the homotopy. That is, $\mathbf{H}(\mathbf{p}_j(t), t) = \mathbf{0}$ for all $t \in (0,1]$ and $1 \leq j \leq D$.

2.2.3. TOTAL-DEGREE HOMOTOPIES. In this subsection, we discuss the famous total-degree homotopy. Total-degree homotopies give us a direct way of constructing good homotopies that were described in §2.2.2. The homotopy has the form:

$$\mathbf{H}(\mathbf{z}, t) = (1-t) \begin{bmatrix} f_1(z_1, \dots, z_N) \\ \vdots \\ f_N(z_1, \dots, z_N) \end{bmatrix} + \gamma t \begin{bmatrix} g_1(z_1, \dots, z_N) \\ \vdots \\ g_N(z_1, \dots, z_N) \end{bmatrix}.$$

Let $d_i = \deg f_i$. Then choose N polynomials g_1, \dots, g_N of degree d_1, \dots, d_N , respectively, so that the system $g_i(\mathbf{z}) = z_i^{d_i} - 1$. We enumerate the start points S_1 as the set:

$$S_1 = \left\{ \left(e^{\frac{2j_1\pi i}{d_1}}, \dots, e^{\frac{2j_N\pi i}{d_N}} \right) \mid 0 \leq j_1 \leq d_1 - 1, \dots, 0 \leq j_N \leq d_N - 1 \right\}.$$

Now construct $\gamma = r \exp(2\theta\pi\sqrt{-1}) \in \mathbb{C}$ sampled randomly. The “gamma trick” ensure that the homotopy is a good homotopy with probability-one [60].

With this construction we call $\mathbf{H}(\mathbf{z}, t)$ a *total-degree homotopy* since the number of solutions of $\mathbf{H}(\mathbf{z}, 1)$ is $D = d_1 \cdots d_N$, the total-degree of the system $\mathbf{f}(\mathbf{z})$. The product of the degrees of a square system $\mathbf{f}(\mathbf{z})$ provides a bound on the number of isolated solutions of $\mathbf{f}(\mathbf{z})$ by applying Bézout’s theorem [69].

2.3. PREDICTOR-CORRECTOR METHODS

In §2.2 we discussed how homotopies are used to find isolated solutions to polynomial systems and defined good homotopies; homotopies with desirable properties for finding solutions. Good homotopies will enable us to design numerical algorithms to approximate solution paths through a homotopy with the ultimate goal of obtaining solutions to polynomials.

2.3.1. CONTINUATION AND PATH TRACKING. Assume we are given a differentiable mapping $\mathbf{p}(t) : (0, 1] \rightarrow \mathbb{C}^N$ where $\mathbf{w} = \mathbf{p}(1)$ is a solution to $\mathbf{H}(\mathbf{z}, 1) = \mathbf{0}$ and $\mathbf{p}(t)$ is a nonsingular solution of $\mathbf{H}(\mathbf{z}, t) = \mathbf{0}$ for $t \in (0, 1]$. Our aim is to compute $\lim_{t \rightarrow 0} \mathbf{p}(t)$ denoted \mathbf{s}_0 . $\mathbf{p}(t)$ has the property that $\mathbf{H}(\mathbf{p}(t), t) = \mathbf{0}$ for $t \in (0, 1]$. This leads to the *Dauidenko differential equation* which places conditions on paths $\mathbf{p}(t)$. Differentiating $\mathbf{H}(\mathbf{z}(t), t) = \mathbf{0}$ with respect to t gives:

$$\frac{\partial \mathbf{H}}{\partial \mathbf{z}} \frac{d\mathbf{z}}{dt} + \frac{\partial \mathbf{H}}{\partial t} = \mathbf{0}.$$

We are not given $\mathbf{p}(t)$ *a priori* but we know that it must satisfy the Dauidenko differential equation. In order to solve the ordinary differential equation (ODE) to find the roots of $\mathbf{H}(\mathbf{z}, 0)$, we first ‘solve’ the ODE for $t \in [\epsilon, 1]$ with $\epsilon \geq 0$, and then estimate \mathbf{s}_0 by approximating the limiting process $\lim_{t \rightarrow 0} \mathbf{p}(t)$. We focus our attention on the region of $t \in [\epsilon, 1]$. *Endgames* are more appropriate for $t \in [0, \epsilon)$ and a full discussion may be found in [69, 9].

2.3.2. PATH TRACKING. We are now in a position to lay the foundation for predictor-corrector methods. Assume we have:

(1) A family of functions on \mathbb{C}^N :

$$\mathbf{H}(\mathbf{z}; \mathbf{q}) = \begin{bmatrix} H_1(z_1, \dots, z_N; q_1, \dots, q_M) \\ \vdots \\ H_N(z_1, \dots, z_N; q_1, \dots, q_M) \end{bmatrix} = \mathbf{0}$$

where H_i are polynomial in $\mathbf{z} \in \mathbb{C}^N$ and complex analytic in $\mathbf{q} \in \mathbb{C}^M$.

(2) Differentiable mappings $\phi : t \in [0, 1] \rightarrow \mathbf{q} \in \mathbb{C}^M$ and $\psi : t \in [0, 1] \rightarrow \mathbf{z} \in \mathbb{C}^N$

where $\mathbf{H}(\psi(t), \phi(t)) = \mathbf{0}$ for $t \in (0, 1]$ and $J\mathbf{H}(\psi(t), \phi(t))$, the Jacobian matrix with respect to \mathbf{z} at $(\mathbf{z}, \mathbf{q}) = (\psi(t), \phi(t))$, has rank N for $t \in (0, 1]$.

We construct \mathbf{H} and ϕ so that ψ exists with $\psi(1) = \mathbf{p}_0$, for some $\mathbf{p}_0 \in \mathbb{C}^N$ (say with a total-degree homotopy). Our goal is to compute $\mathbf{p}^* = \psi(0)$ the target solution corresponding to this path. Differentiating $\mathbf{H}(\psi(t), \phi(t)) = \mathbf{0}$ with respect to t we have the Davidenko initial value problem (IVP):

$$\sum_{i=1}^N \frac{\partial \mathbf{H}(\psi(t), \phi(t))}{\partial \psi_i} \frac{d\psi_i(t)}{dt} + \sum_{i=1}^M \frac{\partial \mathbf{H}(\psi(t), \phi(t))}{\partial \phi_i} \frac{d\phi_i(t)}{dt} = \mathbf{0}, \quad \text{with } \psi(1) = \mathbf{p}_0.$$

Often time we may reduce to the situation when $M = 1$ and $q_1 = t$. This is the case in the total-degree homotopy explain in §2.2.3. Here, after relabeling $\psi(t)$ as $\mathbf{z}(t)$ we then have the Davidenko IVP:

$$\sum_{i=1}^N \frac{\partial \mathbf{H}(\mathbf{z}(t), t)}{\partial z_i} \frac{dz_i(t)}{dt} + \frac{\partial \mathbf{H}(\mathbf{z}(t), t)}{\partial t} = \mathbf{0}, \quad \text{with } \mathbf{z}(1) = \mathbf{p}_0.$$

Let $J\mathbf{H}_\psi(\psi, \phi)$ denote the Jacobian matrix with respect to the variables ψ :

$$J\mathbf{H}_\psi = \frac{\partial \mathbf{H}}{\partial \psi} = \begin{bmatrix} \frac{\partial H_1}{\partial \psi_1} & \cdots & \frac{\partial H_1}{\partial \psi_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial H_N}{\partial \psi_1} & \cdots & \frac{\partial H_N}{\partial \psi_N} \end{bmatrix}$$

evaluated at $(\boldsymbol{\psi}, \boldsymbol{\phi})$ and let:

$$\boldsymbol{\psi}(t) = \begin{bmatrix} \psi_1(t) \\ \vdots \\ \psi_N(t) \end{bmatrix}$$

denote the $N \times 1$ column vector of solutions. Similarly we may arrive at notation for the $N \times M$ matrix $J\mathbf{H}_\phi$ as the Jacobian matrix with respect to the variables $\boldsymbol{\phi}$ and $M \times 1$ column notation for $\boldsymbol{\phi}(t)$. Rewriting the Davidenko differential equation we have:

$$J\mathbf{H}_\phi(\boldsymbol{\psi}(t), \boldsymbol{\phi}(t)) \frac{d\boldsymbol{\phi}(t)}{dt} + J\mathbf{H}_\psi(\boldsymbol{\psi}(t), \boldsymbol{\phi}(t)) \frac{d\boldsymbol{\psi}(t)}{dt} = \mathbf{0}.$$

Since $J\mathbf{H}_\psi(\boldsymbol{\psi}(t), \boldsymbol{\phi}(t))$ is invertible we have:

$$\frac{d\boldsymbol{\psi}(t)}{dt} = -[J\mathbf{H}_\psi(\boldsymbol{\psi}(t), \boldsymbol{\phi}(t))]^{-1} J\mathbf{H}_\phi(\boldsymbol{\psi}(t), \boldsymbol{\phi}(t)) \frac{d\boldsymbol{\phi}(t)}{dt}.$$

Now in the simple case when $M = 1$, and $q_1 = t$ and again relabeling $\boldsymbol{\psi}(t)$ to $\mathbf{z}(t)$, we drop the subscript dependence on $J\mathbf{H}$ and have:

$$\frac{d\mathbf{z}(t)}{dt} = -[J\mathbf{H}(\mathbf{z}(t), t)]^{-1} \frac{\partial \mathbf{H}(\mathbf{z}(t), t)}{\partial t}.$$

There are many approaches to numerically solve the above ODE with given initial condition.

We highlight two special properties of the Davidenko ODE. In the case that $\mathbf{H}(\mathbf{z}, t) = \mathbf{0}$:

- (1) ‘‘Correct’’ the path by using the implicit definition equation $\mathbf{H}(\mathbf{z}, t) = \mathbf{0}$.
- (2) The path is a complex analytic curve so construct a local model of the curve near the endpoint of the path \mathbf{p}^* . This is useful in the region of $t \in [0, \epsilon]$.

2.3.3. PREDICTOR-CORRECTOR METHODS. A class of ODE solvers that work particularly well for implicitly-defined homotopies are called *predictor-corrector methods*. We use the simplified form $\mathbf{H}(\mathbf{z}, t) = \mathbf{0}$ (i.e. when $M = 1$ and $q_1 = t$) as described in §2.3.1. Beginning at $t_0 = 1$ with \mathbf{p}_0 an initial value compute successive approximation $\mathbf{p}_1, \mathbf{p}_2, \dots$ at values

$t_0 > t_1 > \dots > 0$ using *Euler's method* called a *predictor step*:

$$\mathbf{p}_{i+1} = \mathbf{p}_i - [J\mathbf{H}(\mathbf{p}_i, t_i)]^{-1} \frac{\partial \mathbf{H}(\mathbf{p}_i, t_i)}{\partial t} \Delta t_i, \quad \text{where } \Delta t_i = t_{i+1} - t_i.$$

Geometrically, we predict along the line tangent to the solution path centered at (t_i, \mathbf{p}_i) .

Predictors could in theory be used along the entire path. However a more efficient approach is to introduce a *corrector step*. Once a prediction of \mathbf{p}_{i+1} is made using Euler's Method apply *Newton's method* to the polynomial $\mathbf{H}(\mathbf{z}, t_{i+1})$ starting at $\mathbf{z}_0 = \mathbf{p}_{i+1}$:

$$\mathbf{z}_{i+1} = \mathbf{z}_i - [J\mathbf{H}(\mathbf{z}_i, t_{i+1})]^{-1} \mathbf{H}(\mathbf{z}_i, t_i).$$

Here $t = t_{i+1}$ is completely fixed. Replace \mathbf{p}_{i+1} with its updated value after applying Newton's method and then proceed with Euler's method to compute \mathbf{p}_{i+2} and so on. Often times a higher-order predictor such as rk45 is used in order to relax the step size while maintaining a smaller error in prediction.

2.4. PARAMETER HOMOTOPY

This section aims to describe the background for parameter homotopies. Systems of polynomials arising from science and engineering applications almost always have some additional structure that can be exploited during computation. One such notable structure is when a polynomial system is parameterized. That is, several systems of polynomials need to be solved all of which are supported on the same parameter set.

One first solves a parameterized polynomial system at generic parameters using any method such as total-degree (§2.2.3), multihomogeneous (§2.5), or regeneration (§3.3). Then a parameter homotopy solves every subsequent member in the parameterized family using a number of paths equal to the number of nonsingular solutions found in the first step. Often times the number of solutions found in the first step is far fewer than the number of

paths required to find the isolated solutions of each system individually specialized at their parameter values. Thus, parameter homotopies provide an efficient approach to computing a collection of parameterized polynomials.

Let $\mathbf{f}(\mathbf{z}; \mathbf{q}) = \mathbf{0}$ be a parameterized family of polynomial systems:

$$\mathbf{f}(\mathbf{z}; \mathbf{q}) = \begin{bmatrix} f_1(z_1, \dots, z_N; q_1, \dots, q_K) \\ \vdots \\ f_N(z_1, \dots, z_N; q_1, \dots, q_K) \end{bmatrix} = \mathbf{0}.$$

If the system is nonsquare we may “square up” the system using a randomization procedure (§2.1.7). We label $\mathbf{z} \in \mathbb{C}^N$ as the variables and $\mathbf{q} \in \mathbb{C}^K$ are the parameters. There are some fundamental results on parameter homotopies that appear in [61, 69, 9]. That is:

- (1) (Solution preservation) The number of nonsingular isolated solutions of $\mathbf{f}(\mathbf{z}; \mathbf{q}) = \mathbf{0}$ is constant for generic parameter values of $\mathbf{q} \in \mathbb{C}^K$.
- (2) (Path connectedness) Parameters where the number of nonsingular solutions will change from a general set of parameters form a proper algebraic subset of \mathbb{C}^K . In this way this space is a “thin” subset of \mathbb{C}^K . Furthermore, the set of generic parameters form a path connected subset of \mathbb{C}^K .
- (3) (Upper semi-continuity) Specializing at nongeneric parameter values only decreases the number of isolated solutions.
- (4) (Trackable paths) Suppose $\phi(t) \in \mathbb{C}^K$, $t \in [0, 1]$, is a continuous path in parameter space so that the system $\mathbf{f}(\mathbf{z}; \phi(t))$ has the generic number of solutions for $t \in (0, 1]$ (that is, $\phi(t)$ is a generic path). Then:
 - (a) Nonsingular isolated solutions of $\mathbf{f}(\mathbf{z}; \phi(t)) = \mathbf{0}$ vary continuously for $t \in [0, 1]$.
 - (b) The endpoints of solution paths of the homotopy $\mathbf{f}(\mathbf{z}; \phi(t)) = \mathbf{0}$ include all nonsingular isolated solutions of $\mathbf{f}(\mathbf{z}; \phi(0)) = \mathbf{0}$.

The parameter homotopy method is as follows:

- (1) (Ab initio): Select generic parameters $\mathbf{q}^* \in \mathbb{C}^K$ and find all nonsingular solutions \mathcal{S} of $\mathbf{f}(\mathbf{z}; \mathbf{q}^*) = \mathbf{0}$ using any approach.
- (2) (Parameter homotopy) For any $\mathbf{q} \in \mathbb{C}^K$ choose a path $\phi(t) \subset \mathbb{C}^K$ that is generic for $t \in (0, 1]$ with $\phi(1) = \mathbf{q}^*$ and $\phi(0) = \mathbf{q}$. Then follow every nonsingular isolated solution of the homotopy $\mathbf{f}(\mathbf{z}; \phi(t)) = \mathbf{0}$ using solutions \mathcal{S} as $t \rightarrow 0$. The endpoints of these paths include all isolated solutions of $\mathbf{f}(\mathbf{z}; \mathbf{q}^*) = \mathbf{0}$.

In practice, generic parameters \mathbf{q}^* are chosen randomly using a probability-one argument. Genericity and probability-one arguments are discussed in §2.1.6. One can also simplify the choices of generic paths $\phi(t)$. That is, if $\phi(t)$ is chosen as:

$$(1) \quad \phi(t) = t\mathbf{q}^* + (1 - t)\mathbf{q}$$

then $\phi(t)$ will stay generic with probability-one.

There are scenarios where it is beneficial to select \mathbf{q}^* not at random and in this case care must be taken to ensure paths constructed are generic [9]. In addition there is a simplification that can be made if the parameters appear *linearly* in the polynomials.

Instead of constructing $\phi(t)$ explicitly as in equation (1) it suffices to construct the homotopy:

$$\gamma t\mathbf{f}(\mathbf{z}; \mathbf{q}^*) + (1 - t)\mathbf{f}(\mathbf{z}; \mathbf{q}) = \mathbf{0}.$$

For generic choices of $\gamma \in \mathbb{C}$ this homotopy may instead be used at the parameter homotopy stage and may reduce path tracking computations. This homotopy is often useful for implementation purposes especially when there are many parameter values. This parameter homotopy does not work if parameters do not appear linearly.

2.5. MULTIHOMOGENEOUS HOMOTOPY

2.5.1. MULTIPROJECTIVE SPACE. A *multiprojective space* is a product of m projective spaces of various dimensions denoted $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$. When $m = 1$ this becomes the usual projective space \mathbb{P}^{n_1} . One may place homogeneous coordinates on multiprojective space as a cross product of homogeneous coordinates on each individual projective space.

A *multihomogeneous polynomial*:

$$f(z_1, \dots, z_m) : \mathbb{C}^{n_1+1} \times \cdots \times \mathbb{C}^{n_m+1} \rightarrow \mathbb{C}$$

of multidegree (d_1, \dots, d_m) is a polynomial such that:

$$f(\lambda_1 z_1, \dots, \lambda_m z_m) = \lambda_1^{d_1} \cdots \lambda_m^{d_m} f(z_1, \dots, z_m)$$

for every $((\lambda_1, \dots, \lambda_m), z_1, \dots, z_m) \in \mathbb{C}^m \times \mathbb{C}^{n_1+1} \times \cdots \times \mathbb{C}^{n_m+1}$.

EXAMPLE 2.5.1. Consider the polynomial $p(x, y) = xy - 1$. One can first multihomogenize the polynomial by defining homogeneous coordinates $x = X/U$ and $y = Y/W$ so that:

$$P([X, U], [Y, W]) = XY - UW.$$

P is a multihomogeneous polynomial of multidegree $(1, 1)$ defined over $\mathbb{P} \times \mathbb{P}$ since:

$$P(\lambda[X, U], \mu[Y, W]) = (\lambda X)(\mu Y) - (\lambda U)(\mu W) = \lambda\mu(XY - UW) = \lambda\mu P([X, U], [Y, W]).$$

for any $((\lambda, \mu), (X, U), (Y, W)) \in \mathbb{C}^2 \times \mathbb{C}^2 \times \mathbb{C}^2$.

2.5.2. MULTIHOMOGENEOUS HOMOTOPIES. In this section we outline multihomogenous homotopies. Multihomogeneous homotopies and the references within are discussed in [9, 69]. First consider the simple example to motivate the use of multihomogeneous homotopies.

EXAMPLE 2.5.2. Consider the following polynomial system:

$$\mathbf{f}(x, y) = \begin{bmatrix} xy - 1 \\ x^2 - 1 \end{bmatrix} = \mathbf{0}.$$

The total degree of \mathbf{f} is four but one can verify that the system has only two solutions. Put another way we could homogenize \mathbf{f} over \mathbb{P}^2 by defining homogeneous coordinates $x = X/Z$ and $y = Y/Z$ so that:

$$\mathbf{F}_1([X, Y, Z]) = \begin{bmatrix} XY - Z^2 \\ X^2 - Z^2 \end{bmatrix} = \mathbf{0}$$

is a homogenization of \mathbf{f} and has exactly four solutions guaranteed by Bézout's Theorem. Thus, interpreting \mathbf{f} in this context, there are only two solutions of \mathbf{f} over an affine patch ($Z = 1$) of \mathbb{P}^2 .

Now instead, consider multihomogenizing \mathbf{f} over $\mathbb{P} \times \mathbb{P}$. That is, define $x = X/U$ and $y = Y/V$ so that:

$$\mathbf{F}_2([X, U], [Y, V]) = \begin{bmatrix} XY - UV \\ X^2 - U^2 \end{bmatrix} = \mathbf{0}$$

is a multihomogenization of \mathbf{f} . As will be discussed in §2.5.3, \mathbf{F}_2 has only 2 solutions over $\mathbb{P} \times \mathbb{P}$.

As demonstrated in this example, there are advantages in using multihomogenous structures as long as we can define an efficient homotopy method that can be applied over multiprojective space.

Rather than explain multihomogenous homotopies in general, again, we motivate it through example 2.5.2. First notice that:

$$xy - 1 \in \langle xy, x, y, 1 \rangle$$

$$x^2 - 1 \in \langle x^2, x, 1 \rangle$$

(i.e. contained in the ideals) by exploiting the multihomogeneous structure of \mathbf{f} . Then construct a polynomial:

$$\mathbf{g}(x, y) = \begin{bmatrix} (\square x + \square)(\square y + \square) \\ (\square x + \square)(\square x + \square) \end{bmatrix} = \mathbf{0}$$

where ‘ \square ’ denotes generic coefficient parameters. This system has exactly two solutions. As will be explained in §2.5.3 the multihomogeneous root count of \mathbf{f} is two which motivates the use of \mathbf{g} . Using multihomogenous homotopy theory the homotopy:

$$\mathbf{H}(x, y, t) = \gamma t \mathbf{g}(x, y) + (1 - t) \mathbf{f}(x, y) = \mathbf{0}$$

starting at solutions of $\mathbf{g}(x, y) = \mathbf{0}$ are nonsingular for $t \in (0, 1]$ and the endpoints of the paths as $t \rightarrow 0$ include all nonsingular solutions of $\mathbf{f}(x, y) = \mathbf{0}$ for generic choices of $\gamma \in \mathbb{C}$. This homotopy may be extended to multiprojective space $\mathbb{P} \times \mathbb{P}$ by defining the same coordinates as those in example 2.5.2. This produces a homotopy that may be tracked over an affine patch of multiprojective space:

$$\tilde{\mathbf{H}}(X, U, Y, V, t) = \begin{bmatrix} \gamma t(\square X + \square U)(\square Y + \square V) + (1 - t)(XY - UV) \\ \gamma t(\square X + \square)(\square X + \square) + (1 - t)(X^2 - U^2) \\ a_1 X + a_2 U - 1 \\ b_2 Y + b_2 V - 1 \end{bmatrix} = \mathbf{0}$$

for generically chosen a_1, a_2, b_1, b_2 by rescaling the start solutions of $\mathbf{g}(x, y) = \mathbf{0}$ so they satisfy each patch equation.

2.5.3. OBTAINING A MULTIHOMOGENEOUS ROOT COUNT. Obtaining a multihomogeneous root count may be a useful step in performing a multihomogeneous homotopy as discussed above §2.5.2. One may see references [9, 69] for examples and theory on how to compute the root counts. In chapter 4 these computations are used to obtain root counts for the multivariate eigenvalue problem with various multihomogeneous structures.

2.6. NUMERICAL IRREDUCIBLE DECOMPOSITION

The aim of this section is to illustrate how NAG handles positive-dimensional solution sets through the numerical irreducible decomposition (NID). As discussed in §2.1.3 a single polynomial may be decomposed as a union of its irreducible components. In general, the solution set of a polynomial system of equations also decomposes into a union of irreducible components.

Given a polynomial system f_1, \dots, f_n with $X = \mathcal{V}(f_1, \dots, f_n)$ there is a decomposition:

$$X = \bigcup_{i \in \mathcal{I}} X_i$$

where X_i is a pure i -dimensional affine complex algebraic set and \mathcal{I} is a subset of the positive integers $\{0, 1, \dots, \dim X\}$. Furthermore, there are nonempty sets \mathcal{J}_i such that:

$$(2) \quad X = \bigcup_{i \in \mathcal{I}} \bigcup_{j \in \mathcal{J}_i} X_{ij}$$

where each X_{ij} is an irreducible i -dimensional affine complex algebraic set where $X_{ij} \subsetneq \overline{X - X_{ij}}$. A decomposition of X such as equation (2) is called an *irreducible decomposition of X* . The aim of NAG is to develop a numerical analog of the irreducible decomposition called the *numerical irreducible decomposition*. Irreducible components of X will be encoded with a so-called *witness set*.

As discussed in §2.1.3 irreducible components, say X_{ij} , can be written as closures of $X_{ij\text{reg}}$, the set of manifold points of X_{ij} . As stated previously $X_{ij\text{reg}}$ is dense and path connected in X_{ij} .

2.6.1. SLICING AND DEGREE. Every irreducible algebraic set X has a corresponding dimension. One other invariant attached to X is called the *degree*. The degree of X may be found by intersecting X with a general linear space of complimentary dimension to X .

In \mathbb{C}^N , a generic linear space L of dimension k will intersect X (of dimension say m) in an algebraic set of dimension $k + m - N$ [69]. If $k + m - N < 0$ then the spaces will not intersect in a generic sense.

A k -dimensional set in \mathbb{C}^N will have *codimension* $N - k$. Often it is more convenient to phrase statements using codimension rather than dimension. Using the statement about intersection above, if L has codimension m then L will intersect X at isolated points (a 0-dimensional set) with probability-one so as long as L is generically chosen. The parameter space of linear spaces L where this intersection has dimension 0 is dense. Furthermore, the *degree* of X , $\deg X$, is formulated as the number of points of intersection of X with a generic linear space of codimension m . Slicing irreducible sets with linear spaces of complimentary dimension lays the foundation of a witness set and the NID.

2.6.2. WITNESS SETS AND THE NID. Consider an algebraic set X with irreducible decomposition:

$$(3) \quad X = \bigcup_{i \in \mathcal{I}} \bigcup_{j \in \mathcal{J}_i} X_{ij}.$$

A *witness set* for X_{ij} is a set:

$$\mathcal{W}_{ij} = \{\mathbf{f}, \mathbf{L}_{ij}, W_{ij}\}$$

where \mathbf{f} is a system such that X_{ij} is an irreducible component of $\mathcal{V}(\mathbf{f})$, \mathbf{L}_{ij} is a system of i generic linear polynomials and W_{ij} , called a *witness point set*, is a collection of points constructed as the intersection of X_{ij} with $\mathcal{V}(\mathbf{L}_{ij})$.

The *numerical irreducible decomposition* \mathcal{W} of X is a formal union of sets:

$$\mathcal{W} = \bigcup_{i \in \mathcal{I}} \bigcup_{j \in \mathcal{J}_i} \mathcal{W}_{ij}$$

where \mathcal{W}_{ij} corresponds to a witness set of the irreducible component X_{ij} of X .

2.6.3. COMPUTING WITNESS SETS. In §2.6.2 we defined a witness set for an irreducible algebraic set and defined the NID; the numerical analog of the irreducible decomposition.

In order to compute witness sets, for each dimension i one:

- (1) computes a witness point superset \widehat{W}_i in dimension i , and
- (2) reduces a witness point super set \widehat{W}_i to witness sets W_{ij} .

A *witness point superset* is a finite collection of points $\widehat{W}_i \subset \mathcal{V}(\mathbf{f})$ that contains W_{ij} for each i -dimensional irreducible component of $\mathcal{V}(\mathbf{f})$.

Witness Point Supersets. One approach is to construct a *witness point superset* which is constructed one dimension at a time. Although this method is not the state of the art its steps describe the process of computing a witness point superset well and lay the foundation for more sophisticated methods. If X is any algebraic set of pure-dimension k and L is a linear space defined by m generic linear equations then:

- (1) if $m < k$, then $X \cap L$ has dimension $k - m$,
- (2) if $m = k$, then $X \cap L$ has dimension 0 and consists of the $\deg X$ number of isolated points, and
- (3) if $m > k$, then $X \cap L = \emptyset$.

Using the argument explained in section 9.2 of [37] the dimension of each irreducible component of $\mathcal{V}(\mathbf{f})$ is between $N - \text{rank } \mathbf{f}$ and $N - 1$. Definition of rank can be found in §3.2.

Next consider the intersection of $\mathcal{V}(\mathbf{f})$ with $N - \text{rank } \mathbf{f} \leq k \leq N - 1$ hyperplanes and computing isolated points. Consider then the system:

$$\mathbf{g}_i(\mathbf{z}) = \begin{bmatrix} \mathbf{f}(\mathbf{z}) \\ \ell_1(\mathbf{z}) \\ \vdots \\ \ell_{N-1}(\mathbf{z}) \end{bmatrix}$$

for $i = 1, \dots, \text{rank } \mathbf{f}$. This system consists of the intersection of $\mathcal{V}(\mathbf{f})$ with a generic linear space of dimension i using $N - i$ hyperplanes. By counting dimensions $\mathcal{V}(\mathbf{g}_i)$ will consist of isolated points on each $(N - i)$ -dimensional component of $\mathcal{V}(\mathbf{f})$. It may also contain nonisolated points of each component of $\mathcal{V}(\mathbf{f})$ whose dimension is larger than $N - i$. Since the system \mathbf{g}_i may be nonsquare we may require “squaring up” the system. Squaring up and randomization is discussed in §2.1.7. For each $i = 1, \dots, \text{rank } \mathbf{f}$ we solve the system $\mathbf{g}_i(\mathbf{z}) = \mathbf{0}$ and produce a witness point superset \widehat{W}_i that each contain the witness point sets W_{ij} .

Junk Removal. After we have obtained a witness point superset \widehat{W}_i in each dimension i we would like to break these into witness point sets W_{ij} for each irreducible component X_{ij} . Junk removal aims to remove any points in \widehat{W}_i that are contained on components of dimension larger than i called *junk points*.

In fact, removing junk points produces a union of witness point sets W_i for the pure i -dimensional components of $\mathcal{V}(\mathbf{f})$. Note however we are not done at this stage as W_i needs to be further broken into witness point sets W_{ij} for each irreducible component X_{ij} to complete the NID.

Junk points are removed using a combination of the *local dimension test* and the *membership test*. In essence, junk removal using membership testing requires a specialized homotopy. Junk removal is a technical step and we refer the interested reader to [37] for more information.

2.6.4. PURE-DIMENSIONAL DECOMPOSITION. Once the junk points are removed from \widehat{W}_i we obtain W_i ; a union of witness point sets for the pure i -dimensional components of $\mathcal{V}(\mathbf{f})$. This is a capstone computation and finishes a major portion of the NID.

W_i must then be further processed into witness point sets W_{ij} for each irreducible component X_{ij} of $\mathcal{V}(\mathbf{f})$. The subdivision is achieved using a combination of the *trace test* and *monodromy* as described in the following two subsections.

Trace Test. Suppose $X \subset \mathbb{C}^N$ is a pure k -dimensional algebraic set and L is a generic linear space of codimension k . As stated previously $X \cap L$ consists of $\deg X$ number of isolated points. The idea behind a trace test is that the centroid (i.e. the average of these points) will move linearly as the linear space L is moved parallel to itself. Furthermore, if X was instead reducible, then the centroid of some nonempty proper subset of these points would also move linearly as L is moved parallel to itself.

We may detect if the centroid moves linearly by constructing a generic linear combination of each point and then moving the slice L in parallel to two other locations. Using these three test points, we can construct two “slopes”. If the centroid was in fact moving linearly, then the difference of these slopes should be nearly zero. The difference of these slopes is the so-called *trace test of a point*. If we did this for every point in $X \cap L$ considered, then the sum of the various trace tests should evaluate to zero.

Then one applies the trace tests to witness sets as follows. With probability one a set $T \subset W_i$ is a witness point set for an irreducible component if and only if the sum of the traces is zero and is not equal to zero for any nonempty proper subset $S \subset T$.

In practice witness sets for irreducible components may be built by considering all combinations of points from a pure i -dimensional witness set, evaluating their traces, and then seeing what sums of traces evaluate to zero. The trace test method is very computationally intensive as considering a large quantity of points is often too costly. This is especially true for irreducible components with large degrees.

Monodromy. As discussed in the prior subsection on trace test the trace test approach may be very costly especially when the degree of the irreducible components is large. The monodromy method aims to reduce the cost of computing traces by partially breaking points into groups. This partial break up helps alleviate the cost of evaluating so many combinations of traces.

Monodromy is built on the foundation that the set of smooth points of an irreducible component are path connected. If two points lie on an irreducible component X there is a path in X_{reg} between the two points. In addition, the so-called *monodromy action* will detect if two (or more) points lie on the same irreducible component. Detecting a monodromy action involves tracking a collection of homotopies. There is no guarantee that a finite collection of random “monodromy loops” will determine how all the points on pure i -dimensional components are interconnected but this at least aids in computing combination of traces. For more on the specifics of monodromy see [37, 69].

2.6.5. COMPLETING THE PICTURE. Given an algebraic set $X = \mathcal{V}(\mathbf{f}) \subset \mathbb{C}^N$ we summarize computing a NID for $\mathcal{V}(\mathbf{f})$ as follows:

- (1) Compute $\text{rank } \mathbf{f}$. Then for each $N - \text{rank } \mathbf{f} \leq i \leq N - 1$ compute a witness point superset \widehat{W}_i at dimension i by solving $\mathbf{g}_i(\mathbf{z}) = \mathbf{0}$.
- (2) For each witness point superset \widehat{W}_i remove junk points using a combination of membership testing and the local dimension test. This produces W_i ; a union of all the witness point sets W_{ij} .
- (3) Using W_i decompose this further into witness point sets using a combination of the trace test and monodromy.
- (4) The output is W_{ij} a witness point set for each irreducible component X_{ij} of X . This completes the construction of \mathcal{W} ; the NID of X .

CHAPTER 3

PERTURBED REGENERATION

3.1. INTRODUCTION AND MOTIVATION

The field of numerical algebraic geometry (NAG) has a wide reaching set of numerical algorithms which seek to compute and manipulate numerical approximations to systems of polynomial equations². A class of methods called *regeneration methods* [38, 39] seek to compute all nonsingular isolated solutions to a polynomial system by *regenerating* one equation at a time. This will often increase the number of paths required to compute all nonsingular isolated solutions but at the same time increase the quality or conditioning of each path. This reduces the dependence on increased precision and decreased step sizes in adaptive precision which may result in a net increase in performance.

The one disadvantage of regeneration methods is that the multiplicity of isolated singular solutions may be destroyed or in many cases isolated singular solutions may not be obtain at all. *Perturbed regeneration* seeks to broaded the application of regeneration by computing all singular isolated solutions and when the system is *a priori* a square system, we also are capable of correctly computing the multiplicity of each isolated solution using this method.

In this chapter we will compute every isolated solution of $\mathcal{V}(\mathbf{f})$ including isolated singular solutions using homotopy continuation and NAG. We first explain the necessary mathematical background needed (§3.2) and then explain regeneration (§3.3); the main computational

²The aim of this work is to determine the utility of perturbation and regeneration for computing isolated solutions to polynomials. This chapter includes a version of the published manuscript, Perturbed regeneration for finding all isolated solutions of polynomial systems (Daniel J. Bates, Brent R. Davis, David Eklund, Eric Hanson, Chris Peterson, Applied Mathematics and Computation, 2014). My contributions to this publications include organizing the theoretical foundations of the paper, performing runs and analysis for all examples, and experimentation and writing portions of the manuscript. Some sections have been modified from the manuscript so that the thesis as a whole has been presented in a uniform way to adhere to the formatting guidelines.

task we will analyze. Then we discuss the need for a perturbation and why standard regeneration is not sufficient to find all isolated solutions (§3.4). Finally we arrive at the perturbed regeneration algorithm (§3.5) and include various theorems that justify perturbed regeneration so as to ensure all isolated solutions will be computed and that paths preserve multiplicity information (§3.6). Various examples and experiments are considered to evaluate the performance of perturbed regeneration (§3.7) and we discuss related singular homotopy methods and how they compare to perturbed regeneration (§3.8).

3.2. MATHEMATICAL BACKGROUND

Denote the isolated solution of $\mathcal{V}(\mathbf{f})$ as $\mathcal{V}_0(\mathbf{f})$. A necessary but not sufficient condition for a system to contain isolated solutions is that $\text{rank } \mathbf{f}$ is maximal. In what follows we assume this is the case. When the rank is not maximal we may use the numerical irreducible decomposition (NID) to create numerical representatives of irreducible components. See §2.6 for more on the NID.

Let $\mathbf{f} := (f_1, \dots, f_N) : \mathbb{C}^N \rightarrow \mathbb{C}^N$ be a *square* polynomial system with solution set:

$$\mathcal{V}(\mathbf{f}) := \{\mathbf{z} \in \mathbb{C}^N : f_i(\mathbf{z}) = 0, \quad \text{for } i = 1, \dots, N\}.$$

Since $\text{rank } \mathbf{f} = N$ there is a nonempty Zariski open dense subset $\mathcal{Z} \subseteq \mathbb{C}^N$ such that for every $\mathbf{z}^* \in \mathcal{Z}$, $\text{rank } J_{\mathbf{f}}|_{\mathbf{z}=\mathbf{z}^*}$, the rank of the Jacobian matrix of \mathbf{f} at \mathbf{z}^* , equals N .

3.3. REGENERATION

This section outlines the regeneration method to compute all isolated nonsingular solutions of a polynomial system. Let $\mathbf{f} := (f_1, \dots, f_N) : \mathbb{C}^N \rightarrow \mathbb{C}^N$ be a square system with $\text{rank } \mathbf{f} = N$. Denote d_i as the degree of the polynomial f_i for $i = 1, \dots, N$ and:

$$L_i^{(j)}(\mathbf{z}) = \square z_1 + \dots + \square z_N - 1$$

be a linear polynomial with random coefficients for $i = 2, \dots, N$ and $j = 1, \dots, d_i$. Assume the solutions to:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

is solved by reducing the system to a degree d_1 univariate polynomial. Consider the sequence of $d_2 - 1$ homotopies:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(1)}(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(2)}(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(d_2)}(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}.$$

where equations 1, 3, \dots , N are held fixed and at the k^{th} step the 2^{nd} homotopy equation becomes:

$$L_2^{(k)}(\mathbf{z})(1 - t) + L_2^{(k+1)}(\mathbf{z})t$$

for $k = 1, \dots, d_2 - 1$. Denote $S_{2,1}, \dots, S_{2,d_2}$ the set of nonsingular isolated solutions at each step of the sequence of homotopies above. By observation, $\bigcup_j S_{2,j}$ is the set of nonsingular isolated solutions of the system:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(1)}(\mathbf{z})L_2^{(2)}(\mathbf{z}) \cdots L_2^{(d_2)}(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}.$$

The 2nd stage of regeneration is complete by following the homotopy:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ L_2^{(1)}(\mathbf{z})L_2^{(2)}(\mathbf{z}) \cdots L_2^{(d_2)}(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ L_3^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}$$

where the homotopy for equations 1, 3, ..., N are held fixed and the the 2nd homotopy equation is:

$$L_2^{(1)}(\mathbf{z})L_2^{(2)}(\mathbf{z}) \cdots L_2^{(d_2)}(\mathbf{z})(1-t) + f_2(\mathbf{z})t.$$

We say that f_2 has been *regenerated*.

In general the ℓ^{th} step is as follows. Assume that $f_1, \dots, f_{\ell-1}$ have been regenerated.

That is, the nonsingular isolated solutions to:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}$$

are known. Consider the sequence of $d_\ell - 1$ homotopies:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(1)}(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(2)}(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \dots \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(d_\ell)}(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}$$

where the homotopy equations $1, \dots, \ell - 1, \ell + 1, \dots, N$ are fixed and the ℓ^{th} homotopy equation at the k^{th} step becomes:

$$L_\ell^{(k)}(\mathbf{z})(1 - t) + L_\ell^{(k+1)}(\mathbf{z})t$$

for $k = 1, \dots, d_j - 1$.

Let $S_{\ell,1}, \dots, S_{\ell,d_\ell}$ be the set of nonsingular isolated solutions to the d_ℓ systems above.

Then $\bigcup_j S_{\ell,j}$ is the set of all nonsingular isolated solutions of the system:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(1)}(\mathbf{z})L_\ell^{(2)}(\mathbf{z}) \cdots L_\ell^{(d_\ell)}(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}$$

The ℓ^{th} stage of regeneration is complete by following the homotopy:

$$\begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ L_\ell^{(1)}(\mathbf{z})L_\ell^{(2)}(\mathbf{z}) \cdots L_\ell^{(d_\ell)}(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix} \rightarrow \begin{bmatrix} f_1(\mathbf{z}) \\ f_2(\mathbf{z}) \\ \vdots \\ f_{\ell-1}(\mathbf{z}) \\ f_\ell(\mathbf{z}) \\ L_{\ell+1}^{(1)}(\mathbf{z}) \\ \vdots \\ L_N^{(1)}(\mathbf{z}) \end{bmatrix}$$

where the homotopy equations $1, \dots, \ell - 1, \ell + 1, \dots, N$ are held fixed and the ℓ^{th} homotopy equation becomes:

$$L_\ell^{(1)}(\mathbf{z})L_\ell^{(2)}(\mathbf{z}) \cdots L_\ell^{(d_\ell)}(\mathbf{z})(1 - t) + f_\ell(\mathbf{z})t$$

We say that f_ℓ has been regenerated. At every stage of regeneration the nonsingular solutions of the target system are guaranteed with probability one. In addition, all nonsingular solutions to the system that regenerates f_ℓ are guaranteed with probability one.

Once the final polynomial f_N has been regenerated we received a set of solutions $S \subseteq \mathbf{f}(z)$ that contains the set of all nonsingular isolated solutions to $\mathcal{V}(\mathbf{f})$. Additional details and analysis of regeneration are found in the hallmark papers [38, 39].

3.4. NEED FOR PERTURBATION

In this section we illustrate the issues that regeneration has with computing every isolated solution of $\mathcal{V}(\mathbf{f})$. Consider the polynomial system of equations, $\mathbf{f}(x, y) = \mathbf{0}$:

$$\begin{bmatrix} y(x-2)^2 \\ x(y-3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Here, $\mathcal{V}(\mathbf{f}) = \{(0, 0), (2, 3)\}$. The solution $(0, 0)$ is nonsingular and the solution $(2, 3)$ is singular with multiplicity two. Consider the 1st stage of regeneration by solving:

$$\begin{bmatrix} y(x-2)^2 \\ r_1x + r_2y - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

where r_1, r_2 are generic parameters. By implicit substitution the solutions are:

$$\{(1/r_1, 0), (2, -(2r_1 - 1)/r_2)\}$$

where $(2, -(2r_1 - 1)/r_2)$ is a singular solution. At this stage, to apply regeneration directly the isolated solution $(2, -(2r_1 - 1)/r_2)$ must be discarded if we are to regenerate the polynomial $x(y - 3)$. Proceeding with the 2nd stage of regeneration we follow the homotopy:

$$\begin{bmatrix} y(x-2)^2 \\ r_1x + r_2y - 1 \end{bmatrix} \rightarrow \begin{bmatrix} y(x-2)^2 \\ s_1x + s_2y - 1 \end{bmatrix}$$

for generic parameters $s_1, s_2 \in \mathbb{C}$. The polynomial $x(y - 3)$ is regenerated by following the homotopy:

$$\begin{bmatrix} y(x - 2)^2 \\ (r_1x + r_2y - 1)(s_1x + s_2y - 1) \end{bmatrix} \rightarrow \begin{bmatrix} y(x - 2)^2 \\ x(y - 3) \end{bmatrix}.$$

Using this regeneration scheme only the nonsingular solution $(0, 0)$ is obtained.

3.5. PERTURBED REGENERATION ALGORITHM

In this section we will produce an algorithm that computes every isolated solution of $\mathcal{V}(\mathbf{f})$ including isolated singular solutions using homotopy continuation and NAG. Denote each isolated solution to $\mathcal{V}(\mathbf{f})$ as $\mathcal{V}_0(\mathbf{f})$. A necessary but not sufficient condition for a system to contain isolated solutions is that $\text{rank } \mathbf{f}$ is full. We make the additional assumption that $\text{rank } \mathbf{f}$ is full.

To remedy the issue of only guarantees to nonsingular solutions of $\mathbf{f}(\mathbf{z})$ using regeneration directly we introduce a perturbation of the problem. That is, $\mathbf{f}(\mathbf{z})$ is replaced by a polynomial system $\mathbf{f}_p(\mathbf{z}) = \mathbf{f}(\mathbf{z}) - \mathbf{p}$ for a generically chosen parameter $\mathbf{p} \in \mathbb{C}^N$. This trivial perturbation of $\mathbf{f}(\mathbf{z})$ significantly alters the multiplicity structure of $\mathbf{f}(\mathbf{z})$. The perturbed regeneration algorithm is contained in algorithm 1.

Algorithm 1 Perturbed homotopy algorithm

Input: Polynomial system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^N$.

Output: All isolated solutions \mathcal{V} of $\mathbf{f}(\mathbf{z})$.

- 1: Compute $\text{rank } \mathbf{f}$. If $\text{rank } \mathbf{f} < N$, then $\mathcal{V} = \emptyset$.
 - 2: Otherwise, choose a random $p \in \mathbb{C}$. Set $\mathbf{p}^T = (p, p, \dots, p) \in \mathbb{C}^N$.
 - 3: Use a homotopy method (e.g. regeneration, total degree, or multihomogeneous) to numerically approximate all isolated nonsingular solutions T of $\mathbf{f}_p(\mathbf{z}) = \mathbf{f}(\mathbf{z}) - \mathbf{p}$.
 - 4: Follow paths beginning at points T of $\mathbf{f}_p(\mathbf{z}) = \mathbf{f}(\mathbf{z}) - \mathbf{p}$ through a parameter homotopy $\mathbf{f}(\mathbf{z}) - t\mathbf{p}$ for $t \in (0, 1]$ storing every solution $\widehat{\mathcal{V}}$.
 - 5: Remove from $\widehat{\mathcal{V}}$ all non-isolated solutions $\mathbf{z} \in \widehat{\mathcal{V}} - \mathcal{V}$ using a local dimension test to produce \mathcal{V} . One may use the local dimension approach as described in [7].
-

3.6. JUSTIFYING ALGORITHM 1

The theory underlying algorithm 1 is largely due to [37, 59, 69]. Application of this theory has been organized in the context of perturbed homotopies for finding all isolated solutions. In this section we provide justification for algorithm 1.

Recall that $\text{rank} \mathbf{f}$ denotes the rank of a polynomial system. Another characterization of $\text{rank} \mathbf{f}$ is the dimension of $\overline{\mathbf{f}(\mathbb{C}^N)} \subseteq \mathbb{C}^N$; the smallest algebraic subset of \mathbb{C}^N that contains $\mathbf{f}(\mathbb{C}^N)$. $\text{rank} \mathbf{f}$ is an upper bound on the codimension of every irreducible component of $\mathcal{V}(\mathbf{f})$ [69]. An immediate result is that a necessary condition for $\mathbf{f}(\mathbf{z})$ to have isolated solution is $\text{rank} \mathbf{f} = N$.

EXAMPLE 3.6.1. Let $\mathbf{f} : \mathbb{C}^3 \rightarrow \mathbb{C}^3$ be a polynomial system. Suppose $\text{rank} \mathbf{f} = 2$. Then every irreducible component of $\mathcal{V}(\mathbf{f})$ has codimension no more than 2. Therefore, $\mathcal{V}(\mathbf{f})$ contains no isolated solutions.

THEOREM 3.6.1. For any polynomial system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^N$, algorithm 1 produces numerical approximations to all isolated solutions of $\mathbf{f}(\mathbf{z})$ with probability one.

Theorem 3.6.1 may be proven by applying the main result from [50]. We provide an alternative approach in this section. There are three lemmas that support theorem 3.6.1, lemmas 3.6.1–3.6.3.

LEMMA 3.6.1. Given a polynomial system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ there is a Zariski open subset $\mathcal{W} \subseteq \mathbf{f}(\mathbb{C}^N)$ such that for every $\mathbf{p} \in \mathcal{W}$ the solution set of $\mathbf{f}(\mathbf{z}) - \mathbf{p}$ consists of smooth, irreducible components of dimension $N - \text{rank} \mathbf{f}$. In the special case that $\text{rank} \mathbf{f} = N$, the solution set of $\mathbf{f}(\mathbf{z}) - \mathbf{p}$ consists of only nonsingular isolated solutions.

LEMMA 3.6.2. Given a polynomial system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^N$, if $\text{rank} \mathbf{f} = N$, then $\overline{\mathbf{f}(\mathbb{C}^N)} = \mathbb{C}^N$.

LEMMA 3.6.3. Given a polynomial system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ suppose that $\text{rank} \mathbf{f} = N$. There exists a Zariski open subset $\mathcal{W} \subseteq \mathbf{f}(\mathbb{C}^N)$ such that for every $\mathbf{p} \in \mathcal{W}$ and every isolated solution $\mathbf{w} \in \mathcal{V}(\mathbf{f})$ there is at least one smooth path $\mathbf{z}(t)$ beginning at a solution of $\mathbf{f}(\mathbf{z}) - \mathbf{p}$ and ending at \mathbf{w} via the homotopy $\mathbf{H}(\mathbf{z}, t) = \mathbf{f}(\mathbf{z}) - \mathbf{p}$ for $t \in (0, 1]$ and the number of paths leading to \mathbf{w} through the homotopy $\mathbf{H}(\mathbf{z}, t)$ equals the multiplicity of \mathbf{w} as a solution of $\mathbf{f}(\mathbf{z}) = \mathbf{0}$.

If the system $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^M$ is overdetermined ($M > N$) then a randomization $\mathcal{R}(\mathbf{f})$ to a square system will destroy multiplicity structure associated to the isolated solutions of \mathbf{f} . Lemma 3.6.3 shows the utility of perturbed regeneration in the case that the system is square since multiplicity information will be preserved.

PROOF OF THEOREM 3.6.1. If $\text{rank} \mathbf{f} < N$, then $\mathcal{V}(\mathbf{f})$ contains no isolated solutions. Assume $\text{rank} \mathbf{f} = N$. According to lemma 3.6.2 \mathbf{f} is a dominant map and therefore $\mathbf{f}(\mathbb{C}^N)$ is dense in \mathbb{C}^N . By lemma 3.6.1, $\overline{\mathcal{W}} = \mathbb{C}^N$ and the closed set $\mathbb{C}^N - \mathcal{W}$ has codimension at least one.

By lemma 3.6.1, given a random $\mathbf{p} \in \mathbb{C}^N$, the polynomial $\mathbf{f}(\mathbf{z}) - \mathbf{p}$ will contain only nonsingular isolated solutions. It is sufficient to choose a random $p \in \mathbb{C}$ and define $\mathbf{p}^T = (p, p, \dots, p) \in \mathbb{C}^N$ because the algebraic set $\mathcal{P} = \{(p, p, \dots, p) \in \mathbb{C}^N : p \in \mathbb{C}\}$ has codimension $N - 1$. Thus, $(\mathbb{C}^N - \mathcal{W}) \cap \mathcal{P}$ has codimension at least N . Therefore, $\mathcal{P} - (\mathbb{C}^N - (\mathcal{W} \cap \mathcal{P}))$ is dense in \mathcal{W} .

Using NAG, numerical approximations to the solution set of $\mathbf{f}(\mathbf{z}) - \mathbf{p}$ may be computed using regeneration, total degree, or multihomogeneous method for example. Lemma 3.6.3

guarantees that for every isolated solution \mathbf{w} of $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ there is a homotopy path beginning from a point in $\mathcal{V}(\mathbf{f}(\mathbf{z}) - \mathbf{p})$ and ending at \mathbf{w} . By following the homotopy $\mathbf{H}(\mathbf{z}, t)$ as defined in lemma 3.6.3, we obtain a superset $\widehat{\mathcal{V}}$ that contains \mathcal{V} . Finally, using a local dimension test \mathcal{V} may be obtained from $\widehat{\mathcal{V}}$. Thus all isolated solutions of $\mathbf{f}(\mathbf{z})$, including singular isolated solutions are obtained by algorithm 1. \square

In algorithm 1 we obtain a superset $\widehat{\mathcal{V}}$ of solution to $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ that contain all isolated solutions \mathcal{V} . In the case that $\widehat{\mathcal{V}} - \mathcal{V} \neq \emptyset$, algorithm 1 also computes numerical approximations to solutions on positive-dimensional components of $\mathcal{V}(\mathbf{f})$ if these components exist. Note that \mathcal{V} may be obtained from $\widehat{\mathcal{V}}$ via a local dimension test. Non-isolated solutions often exist using total degree or multihomogeneous methods.

Generalization of each lemma may be found in appendix A of [69] as a consequence of Sard's theorem. Lemma 3.6.1 is proven as theorem A.6.1 in [69], lemma 3.6.3 is proven as theorem A.6.1 in [69], and lemma 3.6.2 is given as an exercise in [36] and its related result for a pure d -dimensional algebraic subset is in [37] for $d > 0$. In the assumption of lemma 3.6.2 we have the following proof:

PROOF OF LEMMA 3.6.2. Since $\mathcal{V}(\mathbf{f})$ contains a pure 0-dimensional algebraic subset, $\text{rank } \mathbf{f} = N$. \mathbf{f} is full rank and equivalently $\overline{\mathbf{f}(\mathbb{C}^N)} = \mathbb{C}^N$. \square

Now that theorem 3.6.1 has been justified, there are a few extensions of it.

3.6.1. MULTIPLICITY. The multiplicity, $\mu(\mathbf{z}_i)$, of every isolated solution \mathbf{z}_i of $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ may be computed trivially using algorithm 1. Using the definition of multiplicity defined in [69] we have:

COROLLARY 3.6.1. Algorithm 1 produces the multiplicity $\mu(\mathbf{z}_i)$ of every isolated solution \mathbf{z}_i of $\mathcal{V}(\mathbf{f})$.

Corollary 3.6.1 is justified using theorem A.14.1(3) in [69], that each isolated solution z_i will be the endpoint of $\mu(z_i)$ paths beginning at solutions of $\mathcal{V}(\mathbf{f} - \mathbf{p})$.

3.6.2. NONSQUARE SYSTEMS. In the case of nonsquare systems $\mathbf{f} : \mathbb{C}^M \rightarrow \mathbb{C}^N$ with $M \neq N$, algorithm 1 may be applied using a randomization of \mathbf{f} , $\mathcal{R}(\mathbf{f})$. In this case $\mathcal{V}(\mathbf{f}) \subseteq \mathcal{V}(\mathcal{R}(\mathbf{f}))$ and every isolated solution of $\mathcal{V}(\mathbf{f})$ is also an isolated solution of $\mathcal{V}(\mathcal{R}(\mathbf{f}))$. However, if \mathbf{w} is an isolated solution of $\mathcal{V}(\mathbf{f})$ and $\mu(\mathbf{w}) > 1$, then $\mu(\mathbf{w})$ may increase as an isolated solution of $\mathcal{V}(\mathcal{R}(\mathbf{f}))$. Isolated solutions \mathbf{w} that satisfy $\mathcal{V}(\mathcal{R}(\mathbf{f}))$ but not $\mathcal{V}(\mathbf{f})$ may be filter by evaluating $\mathbf{f}(\mathbf{w})$ and determining if $\mathbf{f}(\mathbf{w})$ is numerically nonzero. Thus, algorithm 1 may be applied to nonsquare systems using randomization with the caveat that multiplicities may increase and additional superfluous solutions may be introduced that must be sifted away.

3.7. EXAMPLES AND EXPERIMENTATION

In this section we evaluate the performance of perturbed homotopies. All computational experiments were tested using Bertini v1.4 [8]. All reported timings except for example 3.7.5 were implemented on a 3.2 GHz core of a Dell Precision Workstation with 12 GB of memory. Example 3.7.5 was implemented using 145 2.67 Ghz Xeon 5640 cores with 144 workers.

3.7.1. SIMPLE ILLUSTRATIVE EXAMPLE. Consider the example from §3.4 to illustrated algorithm 1:

$$\mathbf{f}(x, y) = \begin{bmatrix} y(x - 2)^2 \\ x(y - 3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This system had two isolated solutions $\{(0, 0), (2, 3)\}$. A naïve implementation of regeneration will not obtain the solution $(2, 3)$ where $\mu((2, 3)) = 2$. Using a perturbed homotopy

using regeneration, we first solve the perturbed system:

$$\mathbf{f}_{\mathbf{p}}(x, y) = \begin{bmatrix} y(x-2)^2 - p_1 \\ x(y-3) - p_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

using a naïve implementation of regeneration where $\mathbf{p}^T = (p_1, p_2) \in \mathbb{C}^2$ is chosen randomly.

Suppose that:

$$\mathbf{p}^T = (-0.521957 + 0.810510i, -0.0312394 - 0.602051i).$$

Then the perturbed system $\mathbf{f}_{\mathbf{p}}(x, y)$ has three solutions approximated as:

$$\begin{aligned} (x, y) &= (2.2896 - 0.4818i, 3.0399 - 0.2546i), \\ &= (1.6965 + 0.4895i, 2.8885 - 0.3227i), \\ &= (0.0243 + 0.1930i, -0.0901 - 0.2274i). \end{aligned}$$

This is followed by the homotopy:

$$\mathbf{H}(x, y; t) = \begin{bmatrix} y(x-2)^2 - tp_1 \\ x(y-3) - tp_2 \end{bmatrix}$$

that deforms solutions of $\mathbf{f}_{\mathbf{p}}(x, y)$ to solutions of $\mathbf{f}(x, y)$. Two solution paths approach (2, 3) and the other solution path converges to (0, 0) as guaranteed by corollary 3.6.1.

In summary, by using regeneration on a perturbed system followed by a parameter homotopy we were capable of recovering the singular solution not found using a naïve implementation of regeneration.

3.7.2. CPDM5 SYSTEM. In this example we consider the well-known cpdm5 system from the repository of polynomial systems [77]. The cpdm5 system was originally considered in [29]. This system contains five equations in five variables whose solution set is described in table 3.1. The five singular solutions each have multiplicity 11. As expected, a naïve implementation of regeneration does not find any singular solutions for the cpdm5 solution

TABLE 3.1. Basic properties of the cpdm5 solution set.

	Real solutions	Non-real solutions	Total solutions
Non-singular	38	120	158
Singular	5	0	5
Total	43	120	163

set. Timings and paths tracked for regular and perturbed total degree and regeneration methods are provided in tables 3.2–3.3. One interesting takeaway from the results are that timings for the perturbed runs (regeneration and total degree) vary less than those of the unperturbed runs as indicated by the standard deviation in table 3.2. Users may wish to use a naïve implementation of regeneration if singular isolated solutions are not of importance. While most examples in this section show that perturbed regeneration should be used instead to find all isolated solutions, including singular solutions, this example shows that a total degree (or perturbed total degree) homotopy can be faster.

TABLE 3.2. Run times for the cpdm5 system. Each timing is averaged over 100 runs.

	Step 1	Step 2	Total	Std dev
Perturbed regeneration	2.3 sec	1.2 sec	3.6 sec	0.2 sec
Perturbed total degree	0.7 sec	1.2 sec	1.9 sec	0.2 sec
Regeneration	–	–	4.3 sec	0.9 sec
Total degree	–	–	1.9 sec	0.8 sec

TABLE 3.3. Paths tracked for the cpdm5 system.

	Step 1	Step 2	Total
Perturbed regeneration	363 paths	213 paths	576 paths
Perturbed total degree	243 paths	213 paths	456 paths
Regeneration	–	–	363 paths
Total degree	–	–	243 paths

3.7.3. FAIRNESS OF CRAPS GAME. In [62], the fairness of a game of craps was analyzed when a pair of dice were loaded. Using algebraic geometry, this problem of determining fairness became equivalent to finding nonnegative real solutions to the corresponding polynomial

system found in [62]:

$$\begin{aligned}
\sum_{i=1}^6 p_i &= 1 = \sum_{i=1}^6 q_i \\
p_1 q_1 &= \frac{1}{36} = p_6 q_6 \\
p_1 q_2 + p_2 q_1 &= \frac{2}{36} = p_5 q_6 + p_6 q_5 \\
p_1 q_3 + p_2 q_2 + p_3 q_1 &= \frac{3}{36} = p_4 q_6 + p_5 q_5 + p_6 q_4 \\
p_1 q_4 + p_2 q_3 + p_3 q_2 + p_4 q_1 &= \frac{4}{36} = p_3 q_6 p_4 q_5 + p_5 q_4 + p_6 q_3 \\
p_1 q_5 + p_2 q_4 + p_3 q_3 + p_4 q_2 + p_5 q_1 &= \frac{5}{36} = p_2 q_6 + p_3 q_5 + p_4 q_4 + p_5 q_3 + p_6 q_2 \\
p_1 q_6 + p_2 q_5 + p_3 q_4 + p_4 q_3 + p_5 q_2 + p_6 q_1 &= \frac{6}{36}.
\end{aligned}$$

The system of equations contains 12 variables and 13 equations so is overdetermined. Thus, if we randomize the polynomial system we may destroy some of the multiplicity information. In any case, there are 51 solutions \mathbf{z}_i such that $\sum_i \mu(\mathbf{z}_i) = 252$ with 50 isolated singular solutions and one nonsingular solution. A randomization scheme can be significantly simplified by taking a complex combination of 12 fixed polynomials and one other polynomial.

A randomization procedure does not increase the total degree if we take complex multiples of $\sum_{i=1}^6 p_i - 1$ or $\sum_{i=1}^6 q_i - 1$. Set $f = \sum_{i=1}^6 p_i$. This leads to the square polynomial system:

$$\begin{aligned}
\sum_{i=1}^6 q_i + a_1 f &= 1 \\
p_1 q_1 + a_2 f &= \frac{1}{36} = p_6 q_6 + a_3 f \\
p_1 q_2 + p_2 q_1 + a_4 f &= \frac{2}{36} = p_5 q_6 + p_6 q_5 + a_5 f \\
p_1 q_3 + p_2 q_2 + p_3 q_1 + a_6 f &= \frac{3}{36} = p_4 q_6 + p_5 q_5 + p_6 q_4 + a_7 f \\
p_1 q_4 + p_2 q_3 + p_3 q_2 + p_4 q_1 + a_8 f &= \frac{4}{36} = p_3 q_6 p_4 q_5 + p_5 q_4 + p_6 q_3 + a_9 f \\
p_1 q_5 + p_2 q_4 + p_3 q_3 + p_4 q_2 + p_5 q_1 + a_{10} f &= \frac{5}{36} = p_2 q_6 + p_3 q_5 + p_4 q_4 + p_5 q_3 + p_6 q_2 + a_{11} f \\
p_1 q_6 + p_2 q_5 + p_3 q_4 + p_4 q_3 + p_5 q_2 + p_6 q_1 + a_{12} f &= 6/36.
\end{aligned}$$

A naïve implementation of regeneration found no solutions. Timings for several methods are provided in table 3.4 and paths tracked in table 3.5. Note for this example that perturbed 2-homogeneous performed best when compared to several methods.

TABLE 3.4. Run times for the unfair dice system. Each timing is averaged over 10 runs.

	Step 1	Step 2	Total	Std dev
Perturbed regeneration	47.92 sec	5.28 sec	53.20 sec	6.01 sec
Perturbed total degree	28.50 sec	5.13 sec	33.63 sec	7.27 sec
Perturbed 2-hom	20.97 sec	9.39 sec	30.36 sec	6.54 sec
Total degree	–	–	30.97 sec	6.87 sec
2-hom	–	–	43.68 sec	23.36 sec

TABLE 3.5. Paths tracked for unfair dice system. Regeneration paths averaged over 10 runs.

	Step 1	Step 2	Total	Std Deviation
Perturbed regeneration	2587	475	3062	66
Perturbed total degree	2048	504	2552	–
Perturbed 2-hom	924	504	1428	–

3.7.4. BUTCHER PROBLEM. We consider the following system:

$$\mathbf{f} = \begin{bmatrix} zu + yv + tw - w^2 + (1/2)w - 1/2 \\ zu^2 + yv^2 - tw^2 + w^3 + w^2 - (1/3)t + (4/3)w \\ xzv - tw^2 + w^3 - (1/2)tw + w^2 - (1/6)t + (2/3)w \\ zu^3 + yv^3 + tw^3 - w^4 - (3/2)w^3 + tw - (5/2)w^2 - (1/4)w - 1/4 \\ xzuv + tw^3 - w^4 + (1/2)tw^2 - (3/2)w^3 + (1/2)tw - (7/4)w^2 - (3/8)w - (1/8) \\ xzv^2 + tw^3 - w^4 + tw^2 - (3/2)w^3 + (2/3)tw - (7/6)w^2 - (1/12)w - (1/12) \\ -tw^3 + w^4 - tw^2 + (3/2)w^3 - (1/3)tw + (13/12)w^2 + (7/24)w + 1/24 \end{bmatrix}$$

which first appeared in [18]. Computing the NID [9, 69], the solution set consists of 10 irreducible components of various dimensions provided in table 3.6. In this example all isolated solutions are nonsingular.

When a naïve implementation of regeneration is applied only five nonsingular solutions are approximated. If perturbed regeneration is applied there are 11 nonsingular solutions

TABLE 3.6. Summary of irreducible components of the Butcher problem system.

Dimension	Components	Degree
3	3	1
2	2	1
0	5	1

corresponding to the perturbed system. In step two of perturbed regeneration five nonsingular solutions are obtain; two singular solutions are on a 3-dimensional component and the remaining three points diverge to infinity.

In this experiment, the two singular solutions lie approximately on the same 3-dimensional component which do not lie on the intersection of any two components. An implementation of isosingular deflation [40] by Bertini [8] verifies that these are smooth points on this component.

A perturbed or non-perturbed total degree homotopy also finds points on positive-dimensional components but computation time increases because hundreds of singular solutions are approximated on various components.

Table 3.7 shows timings for various methods. In this example, perturbed regeneration performed best even when compared to basic regeneration. Perturbed regeneration does not encounter singular solutions at any point along regeneration. This is in contrast to non-perturbed regeneration where several singular solutions are encountered at each level of the regeneration procedure.

TABLE 3.7. Run times for the Butcher problem. Every timing is averaged over 100 runs, whereas perturbed and non-perturbed total degree is averaged over 50 runs.

Method	Step 1	Step 2	Total	Std dev
Perturbed regeneration	32.4 sec	0.5 sec	32.9 sec	7.5 sec
Perturbed total degree	663.4 sec	0.5 sec	663.8 sec	113.4 sec
Regeneration	–	–	41.0 sec	15.3 sec
Total Degree	–	–	1106.0 sec	158.3 sec
Regenerative Cascade	–	–	117.4 sec	70.1 sec

3.7.5. NINE-POINT FOUR-BAR DESIGN PROBLEM. We consider the nine-point four-bar design problem. The problem formulation and specific details may be found in example 5.5.2 of [9]. The system has eight equations and variables. Its total degree is 5,764,901, 2-homogeneous root count is 4,587,520, and 4-homogeneous root count is 645,120. There are 8,652 nonsingular isolated solutions total and several positive-dimensional components.

TABLE 3.8. Run times for the nine point problem. Each timing is an average over 10 runs.

Method	Computation Time			
	Step 1	Step 2	Total	Std Dev
Perturbed Regeneration	2 h 18 m 19 s	1 m 19 s	2 h 19 m 38 s	42 m 1 s
Perturbed Total Degree	> 6 h	–	> 6 h	–
Perturbed 2-hom	> 6 h	–	> 6 h	–
Perturbed 4-hom	> 6 h	–	> 6 h	–
Regeneration	–	–	46 m 53 s	24 m 12 s
Total Degree	–	–	> 6 h	–
2-hom	–	–	> 6 h	–
4-hom	–	–	> 6 h	–

Table 3.8 tells us that basic regeneration is the fastest followed by perturbed regeneration. All other homotopy strategies were deemed too costly when compared to perturbed regeneration. Positive-dimensional components are ignored through tracking using basic regeneration but are tracked when following a perturbed regeneration. This partially explains the increase in computational cost where the easiness of path tracking is outweighed by the increased number of paths tracked.

In all homotopy methods from table 3.8, the Bertini [8] configuration settings were held fixed. Configuration settings could be modified for each method independently, but then it would be difficult to compare across each method. As a result of this, a very small portion of paths failed, to a varying degree, across the methods. However, there were approximately 290,000 paths tracked for perturbed regeneration compare to 175,000 paths tracked for basic regeneration.

We conclude that perturbed regeneration does not outperform basic regeneration, but this is not too surprising in the case that there we know *a priori* that there are no singular isolated solutions. Perturbed regeneration does have value in cases where singularities are not known and are interesting to the application.

3.8. SINGULAR HOMOTOPY TECHNIQUES

This section describes preexisting techniques that are used to compute all isolated solutions. §3.8.1 describes how regeneration can be paired with deflation to find singular solutions. In addition, we discuss the regenerative cascade (§3.8.2), a positive-dimensional method where as a by-product enough information is retained to compute isolated singular solutions. Finally in §3.8.3 we describe the cheater’s homotopy which is a general perturbation technique to find singular solutions. The cheater’s homotopy was one of the first methods used to solve polynomial systems that involved a perturbation.

3.8.1. REGENERATION WITH DEFLATION. Regeneration can be combined with a deflation procedure to compute isolated singular solutions [38]. Deflation replaces a polynomial system $\mathbf{f}(\mathbf{z})$ defined on \mathbb{C}^N with a *deflated* polynomial system $\hat{\mathbf{f}}(\mathbf{z}, \boldsymbol{\xi})$ defined on $\mathbb{C}^N \times \mathbb{C}^M$ so that if \mathbf{z}^* is an isolated singular solution to $\mathbf{f}(\mathbf{z})$ then $(\mathbf{z}^*, \boldsymbol{\xi}^*)$ is a *nonsingular* isolated solution of $\hat{\mathbf{f}}(\mathbf{z}, \boldsymbol{\xi})$ [64, 63]. There is a body of work related to deflation highlighting proofs of nonsingularity and strong deflation [49, 63, 40].

In practice, deflation is applied to every intermediate system of regeneration where a singularity may occur. Because the size of the deflated system increases path tracking is less efficient. In addition, deflation often requires a randomization procedure which may destroy the polynomial structure of the equations. Algorithm 1 avoided these type of issues but at the cost of potentially increasing the *relative* paths tracked.

3.8.2. **REGENERATIVE CASCADE.** Regenerative cascade provides an method to compute the NID whose approach is based on regeneration [39]. As compared with basic regeneration, regenerative cascade retains enough information to compute singular isolated solutions. Cascading through dimensions one and a time comes as a significant computational cost if only isolated solutions are of interest. Perturbed regeneration avoids cascading entirely but regenerative cascade is the state of the art if a more complete description of the irreducible components is desired.

3.8.3. **THE CHEATER’S HOMOTOPY.** Parameterized polynomials $\mathbf{f}(\mathbf{z}; \mathbf{p})$ arise frequently in applications where one needs to solve at several parameters $\mathbf{p} = \mathbf{p}_1, \dots, \mathbf{p}_k$ in parameter space. A general discussion of parameter homotopies are discussed in §2.4.

Parameter homotopies first solve a general member in a parameterized family of polynomials $\mathbf{f}(\mathbf{z}, \mathbf{p}^*)$ and then ensure that the solution curves are smooth. The so-called “cheater’s homotopy” addresses this issue by including exactly the same parameter as in lemma 3.6.3 [50].

The cheater’s homotopy solves a parameterized system $\mathbf{f}(\mathbf{z}; \mathbf{p})$ by first solving $\mathbf{f}(\mathbf{z}; \mathbf{p}^*) + \mathbf{q}^*$ where $\mathbf{p}^*, \mathbf{q}^*$ are generically chosen. Solutions to this system are then used as start points as the solutions of the homotopy $\mathbf{H}(\mathbf{z}, t) = \mathbf{f}(\mathbf{z}; t\mathbf{p}^* + (1-t)\hat{\mathbf{p}}) + t\mathbf{q}^*$ at $t = 1$ and solutions for a parameter $\hat{\mathbf{p}}$ of interest are recovered as $t \rightarrow 0$.

To distinguish the work in this chapter we highlight that a perturbation was introduced to improve and extend basic regeneration and not to compete with the cheater’s homotopy. However, regeneration is compatible with a cheater’s homotopy if we use regeneration to solve $\mathbf{f}(\mathbf{z}, \mathbf{p}^*) + \mathbf{q}^*$ for $\mathbf{p}^*, \mathbf{q}^*$ generically chosen.

3.9. PERTURBING POSITIVE-DIMENSIONAL COMPONENTS

A natural question is to consider how positive-dimensional irreducible components interact with perturbation. For example, does a non-reduced component “break apart” into several reduced components? Can we use perturbed regeneration to compute a NID? Is there any information about components of perturbed systems that helps understand its corresponding non-perturbed system? §3.9.1 discusses perturbation in obtaining a NID.

3.9.1. FAILURE TO COMPUTE NUMERICAL IRREDUCIBLE DECOMPOSITION. For a general discussion of NID see §2.6. It is tempting to see if using a perturbed homotopy would provide enough information to find at least one generic point on every irreducible component. That is, given a polynomial system $\mathbf{f}(\mathbf{z})$, consisting of irreducible components, first solve the perturbed system $\hat{\mathbf{f}}(\mathbf{z})$ for which all irreducible components have been “broken” into points under perturbation and then use a homotopy to recover points on the irreducible components of $\mathbf{f}(\mathbf{z})$.

As $\hat{\mathbf{f}}(\mathbf{z}) \rightarrow \mathbf{f}(\mathbf{z})$ the goal is to obtain at least one point on each irreducible component of $\mathcal{V}(\mathbf{f})$. After this, a post-processing step would identify what points belong to the same irreducible component and if a witness set could be verified using monodromy and the trace test. Monodromy and the trace test are discussed in §2.6.4.

We have positive results if $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^n$ is a polynomial system of rank n , with $N \geq n$ and the dimension of the solution set of $\mathbf{f}(\mathbf{z}) = \mathbf{p}$ is $N - n$ for generic $\mathbf{p} \in \mathbb{C}^n$. By intersecting the algebraic set with $N - n$ generic hyperplanes, we reduce to points for which we may track using a homotopy to solutions of $\mathbf{f}(\mathbf{z})$. We are guaranteed at least one point on each irreducible component of $\mathbf{f} : \mathbb{C}^N \rightarrow \mathbb{C}^n$.

However, from a practical standpoint the point obtained may not contain generic properties on each irreducible component. For example, consider the simple polynomial system with multiplicity structure:

$$\mathbf{f}(x, y) = \begin{bmatrix} x^2 \\ xy \end{bmatrix}.$$

It is easy to see that $\mathbf{f}(x, y)$ has rank two and a perturbation $\hat{\mathbf{f}}(x, y)$ has dimension zero for a generic perturbation parameter. One may verify that the solution set of $\mathbf{f}(x, y)$ consists of the y -axis. A perturbed homotopy is used to find a point on the y -axis. However, the point that is *always* obtained is a non-generic point at the origin $(x, y) = (0, 0)$. This simple example illustrates the issues with perturbation to compute the NID. That is, a point is found on the component but it may not be generic.

Further analysis is done on example 3.7.4 involving the Butcher problem. The solution set consists of five positive-dimensional components but a perturbation of the system yields only two point on the five components. After further investigation these points lie at the intersection of one or more of the components. As we can see, perturbation is not a reliable method to construct a NID but is useful in finding isolated singular solutions.

CHAPTER 4

MAX-LENGTH VECTOR LINE OF BEST FIT

4.1. INTRODUCTION AND MOTIVATION

When confronted with a large data set there is a variety of techniques that can be employed to understand the structure of the data set³. When a data set can be thought of a point in a non-Euclidean geometric space, such as a Grassmann or Stiefel manifold, additional insight can be revealed while working with these manifolds [1, 13, 21, 46, 54, 65, 75, 76, 80].

Using the singular value decomposition (SVD), for example, one can model or capture features of a data set when thought of as a linear subspace expressed as the span of a set of orthonormal vectors. Thinking of data as points on Grassmann manifolds and their related Stiefel manifolds has led to algorithms to represent, classify, or compare data sets [20, 34, 74, 58, 79].

The aim of this chapter is to discuss the clustering problem. Given a collection of data, now thought of as a cluster of points on a Grassmann (Stiefel) manifold, we would like to represent the data cluster via a representative point in that space. In practice, this helps to reduce the cost of classification algorithms and related clustering tasks [10, 23, 25, 26, 44, 71].

Generally speaking, our problem is as follows: Let V be a vector space. Suppose \mathcal{C} be finite collection of linear subspaces such that $C \in V$ for every $C \in \mathcal{C}$. Find a line $\ell \in V$ that best represents \mathcal{C} . There are many possible ways to approach this problem. A common

³The aim of this work is to demonstrate how numerical algebraic geometry can be used to solve clustering subroutine problems in geometric data analysis. This chapter includes a version of the published manuscript, The max-length-vector line of best fit to a collection of vector spaces (Daniel J. Bates, Brent R. Davis, Michael Kirby, Justin Marks, Chris Peterson, Numerical Linear Algebra with Applications, 2015). My contributions to this publications include organizing the theoretical foundations of the paper, performing runs and analysis for examples and experimentation including implementation details and writing portions of the manuscript. Minimal changes of content have been made to adhere to the formatting guidelines. Permission to reproduce this manuscript has been granted by John Wiley and Sons under license #4132640334358. Copyright 2015 John Wiley & Sons, Ltd.

method is to express ℓ as the minimum argument of an optimization problem that depends on the data \mathcal{C} . Specifically, we will discuss the scenario where the objective function is expressed as a sum of cosines of singular values between elements of \mathcal{C} and ℓ . We call this representative the *max-length-vector line of best fit for a collection of subspaces* (MLV line).

§4.2 discusses mathematical background such as Grassmann manifolds, principal angles, and manifold means for data clusters on Grassmannians. Then §4.3 discusses the problem formulation and several equivalent forms. In §4.4 we show how the solution will correspond to solutions of a multivariate eigenvalue problem (MEP). Then §4.5 will discuss techniques that will solve the MEP. Finally in §4.6 we look at several examples including an application to a set of image data acquired from the Pattern Analysis Laboratory (PAL) at Colorado State University (CSU).

4.2. MATHEMATICAL BACKGROUND

4.2.1. THE GRASSMANN MANIFOLD AND ITS REPRESENTATIONS. Let $\text{Gr}(p, n)$ denote the set of all p -dimensional linear subspaces of \mathbb{R}^n . $\text{Gr}(p, n)$ has the structure of a smooth manifold of dimension $p(n - p)$. With this structure refer to $\text{Gr}(p, n)$ as a Grassmann manifold. In order to cluster subspaces we would like to represent elements of $\text{Gr}(p, n)$ using matrices. Given an element of $\text{Gr}(p, n)$ represent it as the column space of a full rank $n \times p$ matrix \mathbf{M} . This representation is not unique. For example, if $\mathbf{A} \in \text{GL}(p, \mathbb{R})$, the set of $p \times p$ invertible matrices over \mathbb{R} , then the column space of \mathbf{M} and \mathbf{MA} are equivalent.

Denote $[\mathbf{M}]$ as the equivalence class of full rank $n \times p$ matrices that have the same column space as \mathbf{M} . With this definition, the set of all equivalence classes of this form can be identified with $\text{Gr}(p, n)$. Given an equivalence class $[\mathbf{M}]$, represented by a full rank $n \times p$ matrix \mathbf{M} , we may construct an orthonormal matrix \mathbf{N} via Gram-Schmidt orthonormalization

whose columns span those of \mathbf{M} . Thus $[\mathbf{M}] = [\mathbf{N}] \in \text{Gr}(p, n)$. Furthermore, \mathbf{M} may be represented by $\mathbf{N}\mathbf{A}$ for any $\mathbf{A} \in O(p)$, the space of all $p \times p$ orthonormal matrices. This identifies $\text{Gr}(p, n)$ with the quotient manifold $O(n)/(O(p) \times O(n-p))$. In practice, this identification allows us to represent a data point $[\mathbf{M}] \in \text{Gr}(p, n)$ with an orthonormal matrix. We use this representation throughout the rest of the chapter.

4.2.2. PRINCIPAL ANGLES BETWEEN SUBSPACES. Let $[\mathbf{X}]$ and $[\mathbf{Y}]$ be p and q dimensional nontrivial subspaces of \mathbb{R}^n , respectively. Without loss of generality assume that $p = \dim[\mathbf{X}] \geq \dim[\mathbf{Y}] = q$. The principal angles $\theta_1([\mathbf{X}], [\mathbf{Y}]), \dots, \theta_q([\mathbf{X}], [\mathbf{Y}]) \in [0, \pi/2]$ between $[\mathbf{X}], [\mathbf{Y}]$ are defined recursively as:

$$\begin{aligned} \cos(\theta_k([\mathbf{X}], [\mathbf{Y}])) &= \max_{\mathbf{x} \in [\mathbf{X}]} \max_{\mathbf{y} \in [\mathbf{Y}]} \mathbf{x}^T \mathbf{y} := \mathbf{y}_k^T \mathbf{x}_k \\ \mathbf{x}^T \mathbf{x} &= \mathbf{y}^T \mathbf{y} = 1, \\ \mathbf{y}^T \mathbf{y}_i &= \mathbf{x}^T \mathbf{x}_i = 0, \quad \text{for } 1 \leq i \leq k-1. \end{aligned}$$

The set of orthonormal vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_q\}$ and $\{\mathbf{y}_1, \dots, \mathbf{y}_q\}$ are called the principal vectors for the pair of subspaces $[\mathbf{X}], [\mathbf{Y}]$, respectively. The principal angles have the property that $\theta_k([\mathbf{X}], [\mathbf{Y}]) \leq \theta_{k+1}([\mathbf{X}], [\mathbf{Y}])$ for $1 \leq k \leq q-1$.

There is a nice procedure to compute principal angles [14]. Suppose matrices $\mathbf{Q}_X, \mathbf{Q}_Y$ are orthonormal representatives of the subspaces $[\mathbf{X}], [\mathbf{Y}]$, respectively. Orthonormal representations can be obtained using the QR decomposition. By the singular value decomposition, the $p \times q$ matrix $\mathbf{Q}_X^T \mathbf{Q}_Y$ may be expressed as $\mathbf{U}\Sigma\mathbf{V}^T$ where \mathbf{U} is a $p \times q$ orthonormal matrix, Σ is a $q \times q$ diagonal matrix whose entries are nonnegative real numbers, and \mathbf{V}^T is a $q \times q$ orthogonal matrix. The diagonal entries σ_i of $\mathbf{Q}_X^T \mathbf{Q}_Y$ are called the singular values of $\mathbf{Q}_X^T \mathbf{Q}_Y$ and the columns of \mathbf{U} and \mathbf{V} are called the left-singular vectors and right-singular vectors of $\mathbf{Q}_X^T \mathbf{Q}_Y$. The singular values of $\mathbf{Q}_X^T \mathbf{Q}_Y$ may be ordered so that $\sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_q$.

In [14] there is a strong connection between the principal angles and vectors between $[\mathbf{X}], [\mathbf{Y}]$ and the singular value decomposition of $\mathbf{X}^T \mathbf{Y}$:

$$(4) \quad \text{Principal angles: } \cos \theta_k([\mathbf{X}], [\mathbf{Y}]) = \sigma_k(\mathbf{X}^T \mathbf{Y}) \quad \text{for } 1 \leq k \leq q,$$

$$(5) \quad \text{Principal vectors: } [\mathbf{x}_1 \cdots \mathbf{x}_q] = \mathbf{Q}_X \mathbf{U} \quad \text{and} \quad [\mathbf{y}_1 \cdots \mathbf{y}_q] = \mathbf{Q}_Y \mathbf{V}.$$

Equations (4)–(5) give an efficient way to compute principal angles and principal vectors between subspaces $[\mathbf{X}]$ and $[\mathbf{Y}]$.

4.2.3. KARCHER MEAN. Given a finite collection of points $\mathcal{X} = \{[\mathbf{X}_1], \dots, [\mathbf{X}_k]\} \subset \text{Gr}(p, n)$, define the Karcher mean $[\boldsymbol{\mu}_{\text{KM}}] \in \text{Gr}(p, n)$ for the set \mathcal{X} as:

$$[\boldsymbol{\mu}_{\text{KM}}] = \underset{[\mu] \in \text{Gr}(p, n)}{\text{argmin}} \sum_{i=1}^k \theta_i([\mu], [X_i])^2.$$

The Karcher mean is not unique in general but for points of \mathcal{X} lying close enough to one another the Karcher mean has a unique minimum [10]. In practice, the Karcher mean is approximated iteratively and the approach is guaranteed to converge when the point are close to one another [10].

In addition to the Karcher mean there are several other subspace means and medians that may be placed on a collection of subspaces. In [55], they discuss a variety of subspace means and analyze their quantitative properties on data sets.

4.3. FORMULATIONS OF THE OPTIMIZATION PROBLEM

In this section, after stating the entry optimization problem of interest, several equivalent constructions will be made that will characterize a line as the span of a vector that maximizes the length of a set of unit length vectors each of which are contained in a set of subspaces. Because of this interpretation we call the representative the MLV line.

4.3.1. **PROBLEM FORMULATION.** Let $\mathcal{V} = \{V_1, V_2, \dots, V_k\} \subset \text{Gr}(1, n) \oplus \dots \oplus \text{Gr}(n-1, n)$ and set $d_i = \dim V_i$. Suppose \mathbf{Y}_i is a $n \times d_i$ orthonormal matrix whose column space spans the subspace V_i . From §4.2 we showed that the column space $[\mathbf{Y}_i]$ can be identified with V_i for $1 \leq i \leq k$.

Let $L \in \text{Gr}(1, n)$ and denote $\theta(L, V_i)$ as the principal angle between L and V_i for $1 \leq i \leq k$. In §4.2 we showed an efficient way to compute principal angles between subspaces in general. Define a one-dimensional subspace L_{MLV} as:

$$(6) \quad L_{\text{MLV}} = \operatorname{argmax}_{L \in \text{Gr}(1, n)} \sum_{i=1}^k \cos \theta(L, V_i).$$

Note that L_{MLV} need not be unique. We explain this caveat in §4.6.

In order to introduce data represented as a subspace V_i and put equation (6) in a computable form we must reformulate the corresponding optimization problem. This procedure is summarized in proposition 4.3.1.

PROPOSITION 4.3.1. Suppose $\mathcal{V} = \{V_1, \dots, V_k\} \in \text{Gr}(1, n) \oplus \dots \oplus \text{Gr}(n-1, n)$ and $V_i = [\mathbf{Y}_i]$ where \mathbf{Y}_i is an $n \times d_i$ orthonormal matrix whose columns span V_i with $d_i = \dim V_i$ for $1 \leq i \leq k$. Then L_{MLV} is the span of the longest length vector \mathbf{v} that can be expressed in the form $\mathbf{v} = \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i$ for unit length vectors $\boldsymbol{\alpha}_i \in \mathbb{R}^{d_i}$.

PROOF. Suppose $V \in \text{Gr}(d, n)$ and $V = [\mathbf{Y}]$ for a $n \times d$ orthonormal matrix \mathbf{Y} . According to equation (4), if L is the span of a unit length vector $\boldsymbol{\ell}$, then $\cos \theta(L, V)$ is the singular value of $\boldsymbol{\ell}^T \mathbf{Y}$. Expanding the SVD, we have:

$$\mathbf{Y}^T \boldsymbol{\ell} = \frac{\mathbf{Y}^T \boldsymbol{\ell}}{\|\mathbf{Y}^T \boldsymbol{\ell}\|} \cos \theta(L, V).$$

Therefore $\|\mathbf{Y}^T \boldsymbol{\ell}\| = \cos \theta(L, V)$ but then $\|\operatorname{proj}_V \boldsymbol{\ell}\| = \|\mathbf{Y}^T \boldsymbol{\ell}\|$, thus $\|\operatorname{proj}_V \boldsymbol{\ell}\| = \cos \theta(L, V)$.

By construction, $\operatorname{proj}_V \boldsymbol{\ell}$ minimizes the angle between $\boldsymbol{\ell}$ and any unit length vector $\mathbf{v} \in V$.

Returning to optimization problem (6), we see that:

$$\max_{L \in \text{Gr}(1, n)} \sum_{i=1}^k \cos \theta(L, V_i)$$

is equivalent to:

$$(7) \quad \max_{\boldsymbol{\ell}, \mathbf{v}_i} \sum_{i=1}^k \boldsymbol{\ell}^T \mathbf{v}_i$$

$$(8) \quad \text{subject to } \|\boldsymbol{\ell}\| = 1,$$

$$(9) \quad \mathbf{v}_i \in V_i \text{ with } \|\mathbf{v}_i\| = 1 \text{ for } 1 \leq i \leq k$$

Now if $\mathbf{v}_i \in V_i$ we have $\mathbf{v}_i = \mathbf{Y}_i \boldsymbol{\alpha}_i$ for some $\boldsymbol{\alpha}_i \in \mathbb{R}^{d_i}$. Thus $\|\mathbf{v}_i\| = 1$ implies that $\mathbf{v}_i^T \mathbf{v}_i = \boldsymbol{\alpha}_i^T \mathbf{Y}_i^T \mathbf{Y}_i \boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i = 1$. Therefore equations (7)–(9) are equivalent to:

$$(10) \quad \max_{\boldsymbol{\ell}, \boldsymbol{\alpha}_i} \sum_{i=1}^k \boldsymbol{\ell}^T \mathbf{Y}_i \boldsymbol{\alpha}_i$$

$$(11) \quad \text{subject to } \|\boldsymbol{\ell}\| = 1,$$

$$(12) \quad \|\boldsymbol{\alpha}_i\| = 1 \text{ for } 1 \leq i \leq k.$$

By linearity of matrix multiplication, equations (10)–(12) are equivalent to:

$$(13) \quad \max_{\boldsymbol{\ell}, \boldsymbol{\alpha}_i} \boldsymbol{\ell}^T \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i$$

$$(14) \quad \text{subject to } \|\boldsymbol{\ell}\| = 1,$$

$$(15) \quad \|\boldsymbol{\alpha}_i\| = 1 \text{ for } 1 \leq i \leq k.$$

Set $\mathbf{v} = \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i$. For a fixed $\boldsymbol{\ell}$ since $\|\boldsymbol{\ell}\| = 1$ and $\boldsymbol{\ell}^T \mathbf{v} = \|\boldsymbol{\ell}\| \|\mathbf{v}\| \cos \phi$, where ϕ is the angle between $\boldsymbol{\ell}$ and \mathbf{v} , $\boldsymbol{\ell}^T \mathbf{v}$ is as large as possible when the angle $\phi = 0$. In other words \mathbf{v} is a multiple of $\boldsymbol{\ell}$. Therefore an optimal solution to equations (13)–(15) may be obtained by maximizing the length of \mathbf{v} . Consider then the optimization problem:

$$(16) \quad \max_{\boldsymbol{\alpha}_i} \left\| \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i \right\|^2$$

$$(17) \quad \|\boldsymbol{\alpha}_i\| = 1 \text{ for } 1 \leq i \leq k.$$

Denote the maximum argument to equations (13)–(15) as $\{\boldsymbol{\alpha}_1^*, \dots, \boldsymbol{\alpha}_k^*\}$. By construction $\boldsymbol{\ell}$ points in the direction of $\boldsymbol{v}^* = \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i^*$. Therefore the solution to equation (6) is $L_{\text{MLV}} = [\boldsymbol{v}^*]$. \square

4.3.2. GEOMETRIC INTERPRETATION. In §4.3.1 and proposition 4.3.1 we showed that solving optimization problem (6) was equivalent to optimization problem (13)–(15). The vector $\boldsymbol{v} = \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i$ has a geometric interpretation. The set of column vectors in \mathbf{Y}_i form an orthonormal basis for the subspace V_i . The constraints $\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i = 1$ restrict the vector $\mathbf{Y}_i \boldsymbol{\alpha}_i$ to have unit length. Thus, the vector \boldsymbol{v} is the vector of maximal length that can be obtained by adding k unit length vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_k$ with $\boldsymbol{v}_i \in V_i$. In other words, each vector \boldsymbol{v}_i lies on the $(\dim V_i - 1)$ -dimensional unit hypersphere centered at the origin contained in the respected subspace V_i . With this geometric interpretation we call $L = \text{span } \boldsymbol{v}$ the MLV line.

As mentioned the condition that the columns of \mathbf{Y}_i are orthonormal and $\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i = 1$ shows that $\mathbf{Y}_i \boldsymbol{\alpha}_i$ lies on the unit hypersphere contained in the column space of \mathbf{Y}_i denoted V_i . In applications it is often desirable to relax the condition that \mathbf{Y}_i be instead orthogonal. That is, each column may not necessarily have unit length. By the SVD, a real full-rank orthonormal $n \times k$ matrix \mathbf{Y} may be decomposed as $\mathbf{U}\boldsymbol{\Sigma}$ where \mathbf{U} is an $n \times k$ orthonormal matrix and $\boldsymbol{\Sigma}$ is a $k \times k$ diagonal matrix whose diagonal entries are nonnegative and may be arranged in decreasing order. For some \boldsymbol{v} in the column space of \mathbf{Y} , given an orthogonality condition $\boldsymbol{v}^T \boldsymbol{v} = 1$, we have $\boldsymbol{v} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\alpha}$ for some $\boldsymbol{\alpha}$ so that:

$$\boldsymbol{v}^T \boldsymbol{v} = \boldsymbol{\alpha}^T \boldsymbol{\Sigma} \mathbf{U}^T \mathbf{U} \boldsymbol{\Sigma} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \boldsymbol{\Sigma}^2 \boldsymbol{\alpha} = 1.$$

Since $\boldsymbol{\Sigma}^2$ is clearly positive-semidefinite we may geometrically consider $\boldsymbol{v} = \mathbf{Y}\boldsymbol{\alpha}$ as points on a hyperellipsoid whose semi-axes are defined by the entries of $\boldsymbol{\Sigma}^2$.

Thus, given a collection of orthogonal matrices \mathbf{Y}_i finding the longest vector of the form $\mathbf{v} = \sum_i \mathbf{Y}_i \boldsymbol{\alpha}_i$ subject to $\boldsymbol{\alpha}_i^T \boldsymbol{\Sigma}_i^2 \boldsymbol{\alpha}_i = 1$ is equivalent to finding the longest vector that can be decomposed as a sum of vectors lying in a respective hyperellipsoid contained in the vector space spanned by \mathbf{Y}_i .

4.4. MULTIVARIATE EIGENVALUE PROBLEM

In this section our aim is to solve the nonconvex optimization problem (16)–(17). We provide standard conditions for optimality using Lagrange multipliers and describe iterative solving methods and degenerate cases. In §4.4.1 we show that critical points of (16)–(17) satisfy the so-called multivariate eigenvalue problem (MEP). In §4.4.2 iterative methods to solve the MEP are discussed. Global convergence properties of these methods are discussed in §4.4.3. Finally, in §4.4.4 degenerate cases are discussed which are rare but worth discussing and motivate the methods from NAG.

4.4.1. SOLUTIONS TO THE MLV LINE EQUATIONS. Since the constraints (17) form a convex set and the objective function (16) is continuous the optimal solution to optimization problem (16)–(17) is obtained. The local optimal critical points are solutions of a polynomial system using Lagrange multipliers. Construct the Lagrangian function:

$$\mathcal{L}(\alpha_1, \dots, \alpha_k, \lambda_1, \dots, \lambda_k) = \left\| \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i \right\|^2 - \sum_{i=1}^k \lambda_i (\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i - 1).$$

Our aim is to set the gradient of \mathcal{L} to zero and solve for every solution. For notional convenience denote d_i as the number of columns of \mathbf{Y}_i , and $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \dots, \alpha_{i,d_i})^T$ as an ordered set of coordinates corresponding to V_i with respect to the orthonormal matrix \mathbf{Y}_i . Also set $\boldsymbol{\alpha}^T = (\boldsymbol{\alpha}_1^T, \dots, \boldsymbol{\alpha}_k^T)$ where $\boldsymbol{\alpha}$ is an ordered $(\sum_{i=1}^k d_i) \times 1$ column vector. Employing

a general formulation as in [66], solutions to (16)–(17) satisfy:

$$(18) \quad \nabla_{\boldsymbol{\alpha}} \left\| \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i \right\|^2 + \sum_{i=1}^k \lambda_i \nabla_{\boldsymbol{\alpha}} (\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i - 1) = 0$$

$$\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i - 1 = 0 \text{ for } 1 \leq i \leq k$$

where $\nabla_{\boldsymbol{\alpha}}$ denotes the gradient operator with respect to the variables $\boldsymbol{\alpha}$.

Define the block matrix $\mathbf{Y} = (\mathbf{Y}_1 | \mathbf{Y}_2 | \cdots | \mathbf{Y}_k)$. Since each \mathbf{Y}_i is orthonormal we have the block structure:

$$\mathbf{Y}^T \mathbf{Y} = \begin{pmatrix} \mathbf{I}^{[d_1]} & \mathbf{Y}_1^T \mathbf{Y}_2 & \cdots & \mathbf{Y}_1^T \mathbf{Y}_k \\ \mathbf{Y}_2^T \mathbf{Y}_1 & \mathbf{I}^{[d_2]} & \cdots & \mathbf{Y}_2^T \mathbf{Y}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_k^T \mathbf{Y}_1 & \mathbf{Y}_k^T \mathbf{Y}_2 & \cdots & \mathbf{I}^{[d_k]} \end{pmatrix}$$

where $\mathbf{I}^{[d_i]}$ denotes the $d_i \times d_i$ identity matrix for $1 \leq i \leq k$. The system of polynomials (18) may be written in a more compact form as:

$$(19) \quad \mathbf{Y}^T \mathbf{Y} \boldsymbol{\alpha} = \boldsymbol{\Lambda} \boldsymbol{\alpha}$$

$$\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i = 1 \text{ for } 1 \leq i \leq k.$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1^{d_1}, \lambda_2^{d_2}, \dots, \lambda_k^{d_k})$ is a diagonal matrix that has the multiplier λ_i repeated d_i times for $1 \leq i \leq k$. In the case that \mathbf{Y}_i is relaxed to being simply orthogonal the block diagonal matrices of $\mathbf{Y}^T \mathbf{Y}$ instead become diagonal matrices. We call equations (19) the MLV line equations. In [22, 81] equation (19) is the so-called *multivariate eigenvalue problem* (MEP). An efficient method for solving a MEP based on homotopy continuation is discussed in [22, 81]. We will employ this method using the tools from NAG in §4.5.

4.4.2. ITERATIVE METHODS. We discuss several iterative methods which solve the multivariate eigenvalue problem [22, 43, 81, 82]. The Horst-Jacobi method [43] is a generalization of the power method used to find eigenpairs of an eigenvalue problem. Horst's method applies k steps of the power method to the k block rows of the matrix $\mathbf{Y}^T \mathbf{Y}$ in equation (19).

Expanding on this idea and using the block structure of $\mathbf{Y}^T\mathbf{Y}$ Gauss-Seidel [22] is based on a successive over-relation method.

Local convergence was proven in [22] for Horst-Jacobi and [81] for Gauss-Seidel. Also in [81] a Rayleigh quotient-based algorithm was developed that accounts for the constraints $\boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i = 1$ from problem (16)–(17). The ideas of [22] were then extended to the alternating variable method in [82].

4.4.3. CONVERGENCE TO GLOBAL SOLUTIONS. Nonconvex optimization problems often exhibit the phenomena of multiple local critical points and in some cases have entire critical sets of positive dimension. Often times local methods may either not converge or converge to a local nonglobal critical point. That is, locally it minimizes the objective function but does not globally on the entire feasible set.

In [22] it was shown empirically that Horst-Jacobi often converges to a local *nonglobal* solution. Convergence to a global solution was improved on using a starting point strategy [81]. In the examples presented in [22, 81] it was shown that the Gauss-Seidel algorithm, with a starting point strategy, outperformed Horst-Jacobi. Convergence to a global solution using the alternating variable method outperformed both Horst-Jacobi and Gauss-Seidel [82]. Furthermore in [82] it was shown that global convergence could be further improved based on additional assumptions on the properties of the block diagonal matrices of $\mathbf{Y}^T\mathbf{Y}$ in equation (19). These are applicable in our case where the block diagonal matrices are highly-structured. We demonstrate good global convergence properties in example 4.6.3.

4.4.4. DEGENERATE CASES. In almost every case there is a *unique* MLV line. However, there are many cases where there may be more than one line that minimizes the objective

function in (16). There are several reason when this can occur such as when too few conditions are imposed by the subspaces or when there is a lot of symmetry between subspaces.

Cases where this may occur are as follows:

- (1) (Many finite solutions due to symmetry) There are finitely many solutions. If V_x, V_y, V_z denote the coordinate axes in \mathbb{R}^3 then there are four representative lines of best fit due to the symmetry between the coordinate axes.
- (2) (Infinitely many lines due to dimension) There could be infinitely many lines of best fit due to a positive intersection dimension between subspaces. Since V_1 and V_2 share a subspace and the dimension is larger than one any line in this subspace will maximize the objective function (6).
- (3) (Infinitely many lines due to symmetry) There could be infinitely many lines of best fit due to the symmetry between subspaces. Consider the z -axis and the xy -plane in \mathbb{R}^3 . Due to the symmetry there are an infinitely many number of best fit lines whose union is a pair of cones meeting at the origin; one “above” the xy -axis for $z \geq 0$ and one “below” the xy -axis for $z \leq 0$.

In theory the above cases struggle using the iterative methods described in §4.4.2 since they assume generic behavior such as when there is one and only one line of best fit. The above cases can be handled using a NID using tools from NAG.

4.4.5. MEASURE OF CORRELATION. There are two extreme cases that are worth considering when evaluating the max-length vector. The first case is when k subspaces \mathbf{Y}_i of dimension d_i are mutually orthogonal to one another. In this scenario the matrix $\mathbf{Y}^T \mathbf{Y} = \mathbf{I}_N$

for $N = \sum_{i=1}^k d_i$. Thus, the MLV equation (19) becomes simply:

$$(20) \quad \begin{aligned} \boldsymbol{\alpha} &= \boldsymbol{\Lambda} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i &= 1 \text{ for } 1 \leq i \leq k. \end{aligned}$$

If $\lambda_i \neq 0$ then it must be 1. In either case (the other case being $\lambda_i = 0$) the vector $\boldsymbol{\alpha}_i$ can be any vector so as long as $\|\boldsymbol{\alpha}_i\| = 1$. There are infinitely many solutions to (20) if there is at least one subspace of dimension at least two.

Without loss of generality assume $\boldsymbol{\alpha}_i = \mathbf{e}_1$, the first standard basis vector in \mathbb{R}^{d_i} . Thus the expression $\mathbf{v} = \sum_{i=1}^k \mathbf{Y}_i \boldsymbol{\alpha}_i = \sum_{i=1}^k \mathbf{y}_{i1}$, where \mathbf{y}_{i1} denotes the first column of \mathbf{Y}_i . Furthermore, we know that $\mathbf{y}_{i1} \perp \mathbf{y}_{j1}$ for $i \neq j$ and $1 \leq i, j \leq k$. Since the sum of k orthonormal vectors has length \sqrt{k} the max-length vector has length \sqrt{k} .

The second case is when k subspaces share at least a one-dimensional subspace in common. The max-length vector becomes as long as possible once we select k identical vectors in the intersection of the subspaces V_i . Therefore, the max-length vector has length k since it is the sum of k identical unit-length vectors. If \mathbf{v}^* denotes the max-length vector then $\sqrt{k} \leq \mathbf{v}^* \leq k$.

4.5. A MULTIVARIATE EIGENVALUE HOMOTOPY

As explained in §4.4.2 iterative methods are not guaranteed to converge to the global critical point. Issues of nonconvergence and increasing the probability of reaching a global critical point is discussed in §4.4.3.

It would be beneficial to have a technique that will always find the global solution with probability one using NAG. Using NAG we instead will globally solve the optimization problem using the following procedure:

- (1) Compute all local nonglobal critical points by solving the MLV equation (19).
- (2) For each critical point compute the value of the objective function (16) and from among them select the critical point that maximizes the objective.

We call the homotopy by which we find all critical points the MEV homotopy.

In §4.5.1 we discuss the MEV homotopy which is in many cases optimal for a parameterized family of polynomials. Then in §4.5.2 we compute various root counts for homotopy methods to solve the MLV equations. For a general discussion of homotopy continuation see §2.2. For a general discussion of parameter homotopies see §2.4.

4.5.1. MEV HOMOTOPY. A powerful homotopy method for handling parameterized polynomial systems is a *parameter homotopy*. One first solves a parameterized polynomial system for a general member in the parameter space. One may then solve a special member of the family via a straight-line homotopy through parameter space while tracking solution curves by homotopy continuation. The up-front cost of solving a general member of this family may be costly but each subsequent solve is often significantly more efficient for systems arising from applications.

The MLV equation (19) fit nicely into this framework. In fact a surprisingly simple method will obtain all solutions to a general member *without* using a homotopy. We call this the *multivariate eigenvalue homotopy* (MEV homotopy). The MEV homotopy constructs start solutions similarly to how solutions to a total degree start system may be iterated using analytic expressions for the roots of unity. The homotopy approach was first described in [22] and used curves over real Euclidean space. The following general member is an adequate

start system for a parameter homotopy:

$$(21) \quad \begin{aligned} (\text{diag}(z_1, z_2, \dots, z_N) - \mathbf{\Lambda}) \boldsymbol{\alpha} &= \mathbf{0} \\ \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i - 1 &= 0 \text{ for } 1 \leq i \leq k. \end{aligned}$$

The $z_i \in \mathbb{C}$ are generic random for $1 \leq i \leq N$ and $N = \sum_{i=1}^k d_i$ with notation adopted from §4.4.1.

The start system (21) may be solved directly on each of the k blocks. For the j th block we have the $(d_j + 1) \times (d_j + 1)$ subsystem:

$$(22) \quad \begin{aligned} (\text{diag}(z_1, z_2, \dots, z_{d_j}) - \mathbf{\Lambda}_j) \boldsymbol{\alpha}_j &= \mathbf{0} \\ \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j - 1 &= 0 \end{aligned}$$

where $\mathbf{\Lambda}_j$ is the j th subdiagonal block of $\mathbf{\Lambda}$. The following $2d_j$ solutions to (22) of the form $(\lambda_j, \alpha_{j,1}, \dots, \alpha_{j,d_j})$ are simply the following:

$$\{(z_1, \pm 1, 0, \dots, 0), (z_2, 0, \pm 1, 0, \dots, 0), \dots, (z_{d_j}, 0, \dots, 0, \pm 1)\} \text{ for } 1 \leq i \leq k.$$

Due to the block structure of the solutions we consider solutions to (21) on the produce space $\mathbb{C}^{d_1} \times \mathbb{C}^{d_2} \dots \times \mathbb{C}^{d_k}$ as a k -fold product of the $2d_j$ solutions from every j th block. In total there are $\prod_{j=1}^k 2d_j$ solutions.

Now given a collection of subspaces $V_1, \dots, V_k \in \mathbb{R}^n$, represented by orthonormal matrices $\mathbf{Y}_1, \dots, \mathbf{Y}_k$, we consider its corresponding block matrix \mathbf{Y} and construct the parameter homotopy function:

$$(23) \quad \mathbf{H}(\boldsymbol{\alpha}, \lambda_1, \dots, \lambda_k; t) = \begin{cases} ((t \text{diag}(z_1, z_2, \dots, z_N) + (1-t)\mathbf{Y}^T \mathbf{Y}) - \mathbf{\Lambda}) \boldsymbol{\alpha} \\ \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i - 1 \text{ for } 1 \leq i \leq k. \end{cases}$$

and consider its solutions for $t \in (0, 1]$.

In the limit (as $t \rightarrow 0$) we obtain all isolated solutions of (19) using theory from parameter homotopies. Furthermore, because the cost function in (16) has a natural \pm -symmetry ($\|\mathbf{v}\|^2 = \|-\mathbf{v}\|^2$), we only need solutions up to sign. Put another way (for a fixed t), if

$(\boldsymbol{\alpha}, \lambda_1, \dots, \lambda_k)$ is a solution then $(-\boldsymbol{\alpha}, \lambda_1, \dots, \lambda_k)$ is also a solution to (23). Because the \pm -symmetry persists across the entire homotopy we only track half of the total solutions and reduce the computational cost.

4.5.2. COMPARISON OF ROOT COUNTS. With all continuation methods a general interest is to count the number of solution curves that need to be tracked. Ideally we desire an approach that minimizes the number of paths in tracking to a general member of a family of polynomial systems.

In our case the total degree homotopy requires tracking 2^{N+k} solution curves where $N = \sum_{i=1}^k d_i$. This is easy to count since each polynomial that occurs in (19) has degree two and there are $N + k$ of them.

If we wanted to use a multihomogeneous homotopy with variable grouping $\{\alpha, \lambda\}$ then the first N polynomials have multidegree $(1, 1)$ and the remaining k polynomials have multidegree $(2, 0)$. We obtain the root count using the method mentioned in §2.5.3. That is, construct the expression $(\alpha + \lambda)^N (2\alpha)^k$ and obtain the coefficient on the $\alpha^N \lambda^k$ term. To compute this first find the coefficient of the $\alpha^{N-k} \lambda^k$ term in the expression $(\alpha + \lambda)^N$. The binomial formula of $(\alpha + \lambda)^N$ is:

$$(\alpha + \lambda)^N = \sum_{j=0}^N \binom{N}{j} \alpha^{N-j} \lambda^j$$

so that the coefficient of the $\alpha^{N-k} \lambda^k$ term is $\binom{N}{k}$. The coefficient of the $\alpha^N \lambda^k$ term of $(\alpha + \lambda)^N (2\alpha)^k$ is therefore $\binom{N}{k} 2^k$. Thus, the multihomogeneous root count with variable grouping $\{\boldsymbol{\alpha}, \boldsymbol{\lambda}\}$ is $\binom{N}{k} 2^k$. We summarize the root counts in table 4.1. One remark is that each homotopy method is independent of the dimension of the ambient space \mathbb{R}^n and depends only on the quantity and dimension of the subspaces.

TABLE 4.1. Summary of root counts for various homotopy methods to solve the MLV equations.

Homotopy method	Paths tracked
Total degree	$2^k 2^N$
Multihomogeneous	$2^k \binom{N}{k}$
MEV	$2^{k-1} \prod_{j=1}^k d_j$

4.6. EXAMPLES APPLYING THE MLV LINE

This section highlights a variety of examples illustrating the computations and applications of the MLV line. Examples 4.6.1–4.6.2 illustrate an application of the homotopy continuation method discussed in §4.5.1 on randomly generated data. Example 4.6.3 highlights the iterative alternating variable method (AVM). The AVM was briefly discussed in §§4.4.2–4.4.3. Finally, example 4.6.3 applies the MLV line to a novel data set generated from images taken at the Pattern Analysis Lab (PAL) at Colorado State University (CSU).

4.6.1. **SMALL EXAMPLE.** Consider five randomly generated subspaces represented by full rank matrices $\mathbf{Y}_1, \dots, \mathbf{Y}_5 \in \mathbb{R}^{10}$ of dimensions 4, 3, 3, 2, and 2, respectively. Matrices were generated by sampling each entry uniformly from the interval $[-1, 1]$ and orthonormal matrices were then approximated using a QR decomposition. The MEP homotopy was implemented in parallel with `Bertini v1.3.1` using 18 2.67 GHz Xeon-5650 compute nodes with a CentOS 6.4 operating system. In total 2,304 paths were tracked in approximately 6 seconds. Among the 2,304 paths 1,776 paths converged to finite isolated solutions of which 86 were real. Note that path tracking does not depend on the ambient dimension so this may be increased arbitrarily.

4.6.2. **LARGE EXAMPLE.** Consider nine randomly generated subspaces represented by full rank matrices $\mathbf{Y}_1, \dots, \mathbf{Y}_9 \in \mathbb{R}^{100}$ of dimensions 4, 3, 3, 3, 3, 2, 2, 2, and 2, respectively. Matrices were sampled using the procedure described in example 4.6.1. The MEP homotopy

was implemented in parallel with `Bertini v1.3.1` using 272 2.67 GHz Xeon-5650 compute nodes with a CentOS 6.4 operating system. In total 1,327,104 paths were tracked in approximately 30 minutes. Among the 1.3 million paths only 2,542 solutions were real. In particular the max-length-vector had length 4.27 which is between 3 and 9; the theoretical upper and lower bounds described in §4.4.5.

4.6.3. ITERATIVE METHOD. In this example we consider the performance of the AVM on random orthonormal matrices. 100 sets of five orthonormal matrices of ranks 2 to 10 were considered. In this case random matrices were constructed by choosing entries of the matrix from a standard Gaussian normal distribution then orthonormal matrices were approximated using the QR decomposition. In all cases we embed subspaces in \mathbb{R}^{100} .

In [82], a similar experiment with 1,000 sets of five matrices was designed whose dimensions also varied between 2 and 10. However, matrices considered were not necessarily orthonormal. Success of the method was then quantified using techniques from semidefinite programming. In our case, we instead use the MEP homotopy to find global solutions directly and then use it to assess the performance of the AVM.

In the context of orthonormal matrices the number of iterations required to converge to at least six digits and the success rate of the method in converging to the global optimal solution were improved when compared with table 2 of [82]. The success rate was computed by applying AVM 1,000 times to each set of matrices considered. In all cases we randomly generated a starting point by selecting each entry uniformly.

We measure the ratio of success/failure and then average this ratio across the 100 sets of matrices. Average number of iterations were computed similarly. The average number of iterations was approximately 61 and the success rate of the method was 95% in finding the global solutions on the first attempt.

There is a nonzero probability that the method will fail using just one attempt. However, over multiple attempts the probability will increase dramatically and the iterative method will find the global solution in at least 1,000 trials.

Since all local solutions are found using the MEV homotopy additional information is obtained. For example, the number of local solutions that satisfied the global optimal criterion as in [82]. The global optimal criterion requires that all multivariate eigenvalues to be greater than or equal to one [82]. In this case, there was a relatively small number of solutions that satisfied this condition. We believe the success of AVM relies on the fact that the block diagonal matrices $\mathbf{Y}^T \mathbf{Y}$ are simply identity matrices.

4.6.4. APPLICATION TO IMAGE DATA. In this example we compute the MLV line to a collection of images thought of as subspaces in a high-dimensional ambient space. Images were collected from PAL at CSU. A subset of images were collected from a database consisting of human subjects under varying lightning, illumination, pose, and expression conditions (i.e. smiling, frowning). We limit the scope by considering subjects under varying illumination angles with the ambient lights off under a still neutral expression. The data consists of $1080 \times 1440 \times 3$ arrays which consist of three 1080×1440 arrays separating the red, green, and blue color bands. We then reduce the arrays to matrices by converting to grayscale images. Each matrix is then vectorized column-wise to produce a $1,555,200 \times 1$ column vector.

Three subjects are given labels X , Y , and Z under illuminations sampled across various illumination conditions. Then we select five, six, and seven illuminations of subjects X , Y , Z , respectively. In [11, 12] the set of illuminations of a fixed object is approximated by a convex polyhedral cone. The *illumination cone* can be modeled using a small set of images using a linear subspace [11, 12].

To illustrate the utility of the MLV line we introduce another subject A to X, Y, Z . Then we select three distinct illuminations of A and append these to the illuminations of X, Y, Z . The span of these subspaces have dimensions six, seven, and eight illustrated in figure 4.1. We then change the basis of each subspace by taking random convex combinations with the weight on A reduced shown in figure 4.2. Features of A are nearly hidden after taking



FIGURE 4.1. Rows of pictures correspond to subspaces of subject A together with subjects X, Y, Z under various illuminations.



FIGURE 4.2. Recompute bases for subspaces in figure 4.1. Take convex combinations of generates. Subject A is difficult to identify.

convex combinations. Then compute orthonormal bases for the three subspaces and label them $[\mathbf{Y}_1], [\mathbf{Y}_2], [\mathbf{Y}_3]$ and their matrices $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$, respectively.

The MLV line to $[\mathbf{Y}_1], [\mathbf{Y}_2], [\mathbf{Y}_3]$ is presented in figure 4.3. Two quantitative statistics



FIGURE 4.3. MLV line to $[\mathbf{Y}_1], [\mathbf{Y}_2], [\mathbf{Y}_3]$. We “recover” A although A is “hidden” in figure 4.2

come with the max-length-vector computation including the three multivariate eigenvalues and the length of the max-length-vector. The length of the vector is 2.9416 and the multivariate eigenvalues are:

$$(\lambda_1, \lambda_2, \lambda_3) \approx (2.9087, 2.8673, 2.8770).$$

The length of the max-length vector is a measure of the subspaces “willingness” to share a common line in the optimization sense of problem (16)–(17). The multivariate eigenvalues measure if the solution to problem (16)–(17) may be approximated by the solution of a related eigenvalue problem. That is, if $\lambda_1 = \lambda_2 = \lambda_3$ then the multivariate eigenvalue problem reduces to the standard eigenvalue problem:

$$(24) \quad \mathbf{Y}^T \mathbf{Y} \boldsymbol{\alpha} = \Lambda \boldsymbol{\alpha}$$

$$(25) \quad \boldsymbol{\alpha}^T \boldsymbol{\alpha} = 1.$$

Eigenvalue problem (24)–(25) may be solved numerically with standard methods. The approximate solution to (16)–(17) corresponds to the eigenpair with the largest eigenvalue.

It is interesting to note that the solution to (24)–(25) is related to problem (6) by instead maximizing the sum of the squared cosines of the principal angles [23].

A property we may interpret of the MLV line of best fit is its ability to extract the “most correlated” signal in each subspaces even if the signal is weakly represented. Therefore, the MLV line is more closely tied to the properties of the extrinsic manifold mean, the L_2 -median, and the flag mean [55, 23] rather than the Karcher mean which tries to “average” out the subspaces.

CHAPTER 5

MODEL SELECTION

5.1. INTRODUCTION AND MOTIVATION

There are a variety of scenarios where mathematical models are constructed and studied to better understand real-world phenomena⁴. In some situations several mathematical models are constructed that are built upon alternative hypotheses. The central question then becomes: “What model best explains the experimental data?”. In other words, we would like to select what model best fits noisy experimental data. This is a central problem called *model selection*, a fundamental scientific problem [17, 19, 47].

For example, when dealing with models from the life sciences a standard procedure for model selection is to estimate all model parameters and hidden variables and then select a model with the minimal best-fit error that also minimizes model complexity [51, 52].

For the situations we will consider the models are described as the steady-state equilibrium of polynomial ordinary differential equations (ODEs) whose system of equations are labeled $\mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0}$ where \mathbf{a} and \mathbf{x} denote the model parameters and variables, respectively. As is often the case, some of the variables in \mathbf{x} may not be measurable. However, we may have measurable ‘outputs’ $\mathbf{z} = \mathbf{g}(\mathbf{x})$ that depend on the non-measurable variables in \mathbf{x} .

⁴The aim of this work is to determine how numerical algebraic geometry can be used in model selection. This chapter includes a version of the published manuscript, Numerical algebraic geometry for model selection and its application to the life sciences (Elizabeth Gross, Brent R. Davis, Ken Ho, Daniel J. Bates, Heather Harrington, Journal of the Royal Society Interface, 2016). My contributions to this publications include organizing the theoretical foundations of the paper, performing runs and analysis for examples and experimentation including implementation details and writing portions of the manuscript. The manuscript and its supplementary material have been combined so that the thesis as a whole has been presented in a uniform way to adhere to the formatting guidelines. The manuscript and supplementary material has been published as open access under the CC-BY licence v4.0.

Model selection may then be formulated as a least-squares optimization problem:

$$(26) \quad \min_{\mathbf{a}, \mathbf{x}, \mathbf{z}} \|\mathbf{z} - \mathbf{y}\|^2 \quad \text{s.t.} \quad \begin{cases} \mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0} \\ \mathbf{z} = \mathbf{g}(\mathbf{x}) \end{cases}$$

where \mathbf{y} denotes the observed data. In general, problem (26) is a non-convex optimization problem and there is no guarantee that a local approach [48, 3] will find a global critical point. Since \mathbf{f} and \mathbf{g} are polynomials problem (26) may be solved globally by finding all roots of an associated polynomial system using a Lagrange multiplier method.

The aim of this chapter is to propose a method for model selection using polynomial deterministic models using techniques from NAG. As stated in §2.2 and §2.1.6 a NAG approach has a probability-one guarantee to find all isolated solutions to a polynomial system of equations. This may be interpreted as finding all isolated critical points of problem (26).

In §5.2 model fitting and parameter estimation will be approached using a maximum-likelihood perspective. In §5.3 the geometry of problem (26) will be discussed which will allow us to make sense of dimensions of intersection of the model and data varieties. Finally in §5.5 the NAG approach will be applied to three examples from biology: cell death activation, HIV progression, and multisite phosphorylation using experimental data.

5.2. PROBLEM STATEMENT

Consider a mathematical model whose dynamics are described using a system of first-order polynomial ODEs:

$$(27) \quad \mathbf{x}'(t) = \mathbf{f}(\mathbf{a}, \mathbf{x})$$

where $\mathbf{a} = (a_1, \dots, a_k)$ are parameters (e.g. rate constants in a deterministic model such as a chemical reaction network with mass-action kinetics), $\mathbf{x} = (x_1, \dots, x_n)$ are variables, and $\mathbf{f} = (f_1, \dots, f_r)$ are polynomials in \mathbf{x} and \mathbf{a} with measurable outputs $\mathbf{z} = \mathbf{g}(\mathbf{x})$ where

$\mathbf{z} = (z_1, \dots, z_m)$ ($m \leq n$) and $\mathbf{g} = (g_1, \dots, g_m)$ are polynomials in \mathbf{x} . In our discussion of the problem statement the parameters in \mathbf{a} will be considered as fixed variables but during computational stages they may be grouped together with \mathbf{x} and be also called variables.

Define the *real model variety* is the solution set of the system:

$$(28) \quad \mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0}$$

$$(29) \quad \mathbf{z} - \mathbf{g}(\mathbf{x}) = \mathbf{0}, \quad \text{that is,}$$

$$(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} := \{(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathbb{R}^{k+n+m} : \mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0}, \mathbf{z} - \mathbf{g}(\mathbf{x}) = \mathbf{0}\}$$

corresponding to the steady state equilibria of the model. In the situation that only one data point $\mathbf{y} = (y_1, \dots, y_m)$ is considered the *real data variety* is the affine linear space:

$$(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} := \{(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathbb{R}^{k+n+m} : z_i = y_i, 1 \leq i \leq m,$$

with $\dim(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} = k + n$. As with all experimental data it is important to consider the possibility for extrinsic measurement error. That is, there are errors $\{\epsilon_1, \dots, \epsilon_m\}$ on the observable data \mathbf{y} . Assume the errors are uncorrelated random variables and each error ϵ_i is normally distributed with known variance σ_i . The three fundamental scientific problems we will consider: model validation, model selection, and parameter estimation may then be described using $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$.

5.2.1. MODEL VALIDATION. The fundamental problem of model validation is to determine whether a polynomial model \mathcal{M} is compatible with data according to a certain significance level α . Using the noise assumption described in §5.2 above each model \mathcal{M} gives rise to a *statistical* model.

Given a deterministic system $\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}, \mathbf{a})$ with an observation \mathbf{y} made a steady-state the statistical model under consideration is:

$$(30) \quad Y_i = z_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i), \quad 1 \leq i \leq m$$

$$(31) \quad \mathbf{f}(\mathbf{x}, \mathbf{a}) = \mathbf{0}$$

$$(32) \quad \mathbf{z} - \mathbf{g}(\mathbf{x}) = \mathbf{0}$$

where $\mathbf{x}, \mathbf{a}, \mathbf{z}$ are all unknown, and σ_i is known for all i .

The question of model compatibility may be formulated as asking if a model is a ‘good fit’ for the data using significance testing. A common goodness-of-fit statistic is:

$$(33) \quad d^2 := \min \sum_{i=1}^m \frac{(z_i - Y_i)^2}{\sigma_i^2} \text{ subject to } (\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}.$$

When the variances σ_i^2 differ d^2 may be thought of as the minimum-squared weighted Euclidean distance between $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. Under the assumption that the variances are all one the statistic (33) is the standard minimum-squared distance. In what follows assume all variances are one since we could rescale variables and observable data in the case the variances differ from one (but are equal).

Optimization problem (33) may be interpreted using maximum-likelihood estimation. Assume a data point $\mathbf{y} = (y_1, \dots, y_m)$ is a perturbation $\mathbf{y} = \boldsymbol{\xi} + \boldsymbol{\epsilon}$ of some unknown true value $\boldsymbol{\xi} = (\xi_1, \dots, \xi_m)$, where each component ϵ_i of the error $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_m)$ is an independent zero-mean Gaussian random variable with variance σ_i^2 . The aim is to determine the probability that \mathbf{y} comes from a model defined by $\mathcal{V}_{\mathcal{M}}$. As described above a point on $\mathcal{V}_{\mathcal{M}}$ has the form $(\mathbf{a}, \mathbf{x}, \mathbf{z})$.

The probability that \mathbf{y} comes from a given point $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}}$ (i.e. that \mathbf{y} is a perturbation of \mathbf{z} where $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}}$ for some \mathbf{a} and \mathbf{x}) is then:

$$\Pr(\mathbf{y} | \mathbf{a}, \mathbf{x}, \mathbf{z}) = \Pr(\mathbf{y} | \boldsymbol{\xi} = \mathbf{z}) = \prod_{i=1}^m \Pr(y_i | \xi_i = z_i).$$

This is also called the likelihood $L(\mathbf{a}, \mathbf{x}, \mathbf{z} | \mathbf{y})$ of $(\mathbf{a}, \mathbf{x}, \mathbf{z})$ and the aim is to find its maximizer over all $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}}$. This can equivalently be done by considering the log-likelihood which gives:

$$\log L(\mathbf{a}, \mathbf{x}, \mathbf{z} | \mathbf{y}) = \sum_{i=1}^m \log \Pr(y_i | \xi_i = z_i) = \sum_{i=1}^m \left(\frac{1}{2} \log 2\pi\sigma_i^2 - \frac{(z_i - y_i)^2}{2\sigma_i^2} \right)$$

by the so-called normality assumption. The maximizer $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$ can therefore be found by solving the optimization problem:

$$d^2 = \min_{(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}}} \sum_{i=1}^m \frac{(z_i - y_i)^2}{\sigma_i^2},$$

where the optimum is the test statistic (33). The values $\hat{\mathbf{a}}$, $\hat{\mathbf{x}}$, and $\hat{\mathbf{z}}$ are the maximum likelihood estimates for, respectively, the parameters, the unobservable variables, and the output values.

The test statistic d^2 itself also has a useful interpretation. Suppose that \mathbf{y} comes from a point $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}}$. Then:

$$d^2 = \sum_{i=1}^m \frac{(\hat{z}_i - y_i)^2}{\sigma_i^2} \leq \sum_{i=1}^m \frac{(z_i - Y_i)^2}{\sigma_i^2}$$

by definition of minimum. Regarding each y_i as a random variable each term $(z_i - y_i)/\sigma_i$ in the summation above is standard normal. Therefore, the right-hand side has a chi-squared distribution with m degrees of freedom (χ_m^2). The inequality should be interpreted by regarding d^2 as a random variable subject to the same source of randomness. This can

be written clearer as:

$$d^2(\omega) \leq \sum_{i=1}^m \frac{(z_i - y_i(\omega))^2}{\sigma_i^2},$$

where the underlying dependence of both sides on the same random realization ω is explicitly written and the inequality then holds for each value of ω . Consequently, we conclude that:

$$\Pr(d^2 \leq u) \geq \Pr(U \leq u), \quad U \sim \chi_m^2,$$

so

$$(34) \quad \Pr(d^2 \geq p_\alpha) \leq \Pr(U \geq p_\alpha) = \alpha, \quad U \sim \chi_m^2,$$

where p_α is the upper α -percentile for χ_m^2 . This can be used to test the hypothesis that \mathbf{y} comes from \mathcal{V}_M .

In summary what has been shown is that minimizing the argument in (33) is equivalent to maximizing the log-likelihood function:

$$(35) \quad \log L(\mathbf{a}, \mathbf{x}, \mathbf{z}|\mathbf{y}) = \sum_{i=1}^m \left(\frac{1}{2} \log 2\pi\sigma_i^2 - \frac{(z_i - y_i)^2}{2\sigma_i^2} \right)$$

and in addition provides an approach to test the hypothesis that \mathbf{y} comes from \mathcal{V}_M using inequality (34).

Model compatibility may be summarized as follows. The null hypothesis is that *the observable data \mathbf{y} is generated from the statistical model defined by \mathcal{M}* . As stated above the distribution function of d^2 is dominated by χ_m^2 . We reject the null hypothesis and call model \mathcal{M} *incompatible* if the observed value d^2 is greater than p_α , the upper α -percentile for χ_m^2 . Otherwise, we fail to reject the null hypothesis and say the model \mathcal{M} is *compatible* with significance level α .

There are a few subtle features of model compatibility that must be discussed moving forward. It may be the case that the real model and data varieties intersect, that is

$(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \neq \emptyset$ so that $d^2 = 0$. In this case, we say the model is compatible with the data. Furthermore, if there are restrictions on $(\mathbf{a}, \mathbf{x}, \mathbf{y})$ (i.e. all parameters and variables should be nonnegative) then finding d^2 will become a *constrained* optimization problem. These aspects will be discussed in more detail in §5.3.

5.2.2. MODEL SELECTION. The fundamental problem of model selection is given a set of models, $\{\mathcal{M}_1, \dots, \mathcal{M}_s\}$ one wants to determine the model of best fit to prescribed data. Deciding the model of best fit comes down to selecting a model that minimizes the test statistic (33).

If the test statistic d^2 evaluates to zero for all (or even multiple) models then we are unable to make a selection between models. This issue can be addressed by designing (potentially more costly) experiments that yield more relevant information to help select a model. When more variables are measured the intersection $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ is often reduced. Once this intersection is empty across every model considered model selection can be performed.

5.2.3. PARAMETER ESTIMATION. The fundamental problem of parameter estimation is finding the point $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}}) \in (V_{\mathcal{M}})_{\mathbb{R}}$ that minimizes the test statistic (33). This may be put in a maximum-likelihood estimate context as finding the point $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$ that maximizes the log-likelihood function (35) under prescribed noise assumptions.

The parameters $\hat{\mathbf{a}}$ may be extracted directly from $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$. In addition the estimate on hidden variables $\hat{\mathbf{x}}$ and the denoised outputs $\hat{\mathbf{z}}$ are also found. As described in §5.2.1 there may be concerns about the intersection $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ being nonempty. In this scenario there may be more than one choice for $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$. If the intersection is empty one is selecting points that geometrically minimize the distance between $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. Setting up these polynomial systems is discussed in detail in §5.3.

5.3. GEOMETRY

In §5.2 three fundamental problems of model compatibility, model selection, and parameter estimation were discussed. In all the cases we must evaluate the test statistic d^2 (33) considered as a nonlinear optimization problem. One must also consider the dimension of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ and specifically the situation where the intersection is nonempty. Geometrically one is minimizing the distance between the algebraic varieties $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ or determining if they intersect at real points.

Most optimization methods for solving nonlinear problems are local in nature, that is, there is no guarantee the method will obtain the global minimum. However, using NAG we are able to obtain all local extrema over \mathbb{C} with probability one. Employing NAG to solve nonlinear optimization has been used in other contexts [32, 72] as well as in chapter 4 in computing the MLV line.

It is important to discuss the underlying geometry between $(\mathcal{V}_{\mathcal{M}})$ and $(\mathcal{V}_{\mathcal{D}})$ as this will lay the foundation for computing the test statistic (33). Let $\mathcal{V}_{\mathcal{M}} \subseteq \mathbb{C}^{k+n+m}$ be the (complex) Zariski closure of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $\mathcal{V}_{\mathcal{D}} \subseteq \mathbb{C}^{k+n+m}$ be the (complex) Zariski closure of $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. See §2.1.4 for a discussion on the Zariski closure and its topology. $\mathcal{V}_{\mathcal{M}}$ and $\mathcal{V}_{\mathcal{D}}$ will be called the *model variety* and *data variety*, respectively, and they are distinguished between their real counterparts $(V_{\mathcal{M}})_{\mathbb{R}}$ and $(V_{\mathcal{D}})_{\mathbb{R}}$ which we called the real model and data varieties from §5.2.

The intersection of $\mathcal{V}_{\mathcal{M}}$ and $\mathcal{V}_{\mathcal{D}}$ is comprised of the solution set of the union of polynomials defining $\mathcal{V}_{\mathcal{M}}$ and $\mathcal{V}_{\mathcal{D}}$, respectively. This union may be represented numerically using a witness set via the numerical irreducible decomposition (NID). The background for witness sets and NID is described in §2.6. The intersection may be composed of several irreducible components of varying dimensions. Since we are interested in only the real points we may use the method described in [37] to determine if there are any feasible real points in $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$.

In the case that $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ is empty we are geometrically selecting points on the model variety and data variety that minimize the distance to one another. The set of points may also be defined using a polynomial system of equations. NAG techniques may then be employed to solve the system. A well-known necessary condition for local extrema is given by the Fritz John (FJ) conditions related to Lagrange multipliers. In what follows we assume that $r + m = \text{codim}V_{\mathcal{M}}$; however, when this is not the case the number of equations can be reduced as will be demonstrated in §5.5.

PROPOSITION 5.3.1 (Equations given by Fritz John conditions). Let $r + m = \text{codim} \mathcal{V}_{\mathcal{M}}$. Let $\mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0}$, $\mathbf{z} - \mathbf{g}(\mathbf{x}) = \mathbf{0}$ be defined on a Zariski open set of $\mathcal{V}_{\mathcal{M}}$ and define $\mathbf{h}(\mathbf{a}, \mathbf{x}, \mathbf{z}) = \{\mathbf{f}(\mathbf{a}, \mathbf{x}), \mathbf{z} - \mathbf{g}(\mathbf{x})\}$ (for simplicity of notation below). If $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ is a local minimum of:

$$(36) \quad \sum_{i=1}^m (z_i - y_i)^2,$$

then there exists $\boldsymbol{\lambda} := (\lambda_0, \lambda_1, \dots, \lambda_{r+m}) \in \mathbb{P}^{r+m}$ such that $(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda})$ is a solution to the system:

$$(37) \quad \mathbf{f}(\mathbf{a}, \mathbf{x}) = \mathbf{0},$$

$$(38) \quad \mathbf{z} - \mathbf{g}(\mathbf{x}) = \mathbf{0},$$

$$(39) \quad \lambda_0 \begin{bmatrix} \mathbf{0} \\ \mathbf{z} - \mathbf{y} \end{bmatrix} + \sum_{i=1}^{r+m} \lambda_i \nabla_{\mathbf{a}, \mathbf{x}, \mathbf{z}} h_i(\mathbf{a}, \mathbf{x}, \mathbf{z}) = \mathbf{0},$$

where \mathbb{P}^{r+m} refers to complex projective space and $\nabla_{\mathbf{a}, \mathbf{x}, \mathbf{z}}$ refers to the operator consisting of all first-order derivatives with respect to \mathbf{a} , \mathbf{x} , and \mathbf{z} .

Solving system (37)–(39) using NAG produces all local extrema of the objective (36). From there we select a pair of nearest points that minimize the objective.

5.3.1. NAG TECHNIQUES USED. Much of the NAG background necessary for this chapter is discussed in chapter 2 but it is useful to discuss specific NAG aspects employed in §5.5 with broad strokes.

If $\boldsymbol{x} \in \mathbb{R}^N$ is a *real* solution of $\boldsymbol{f} = \mathbf{0}$ it is either isolated among the complex solutions or it lies on a positive-dimensional complex irreducible component. In the former case the methods of NAG will find \boldsymbol{x} and recognize it as real. In the latter case \boldsymbol{x} can be difficult to uncover.

For the purposes of this chapter it is usually only required to verify the existence of a real solution especially in model compatibility discussed in §5.2.1. In this case we can find witness points on all positive-dimensional components and then use the procedure in §2.1 of [37] to verify the existence of real points.

In addition parameterized homotopies will be employed throughout §5.5 and as described in §2.4 this approach significantly reduces computational complexity when many parameter values are considered.

5.4. ALGORITHMS

In this section we outline three algorithms related to model validation, model selection, and parameter estimation discussed in §5.2. Then §5.4.1 discusses the main algorithm for model validation. In §5.4.2 a detour is taken to consider the case when variable and parameter are nonnegative. Then in §§5.4.3–5.4.4 we return to the two final algorithms to solve the model selection and parameter estimation problem both of which will be build upon algorithm 2 for model validation. Figure 5.1 helps to illustrate the three algorithms considered and a simple illustrative example is discussed in §5.4.5 as a lead in to the more complicated models considered from the life sciences in §5.5.

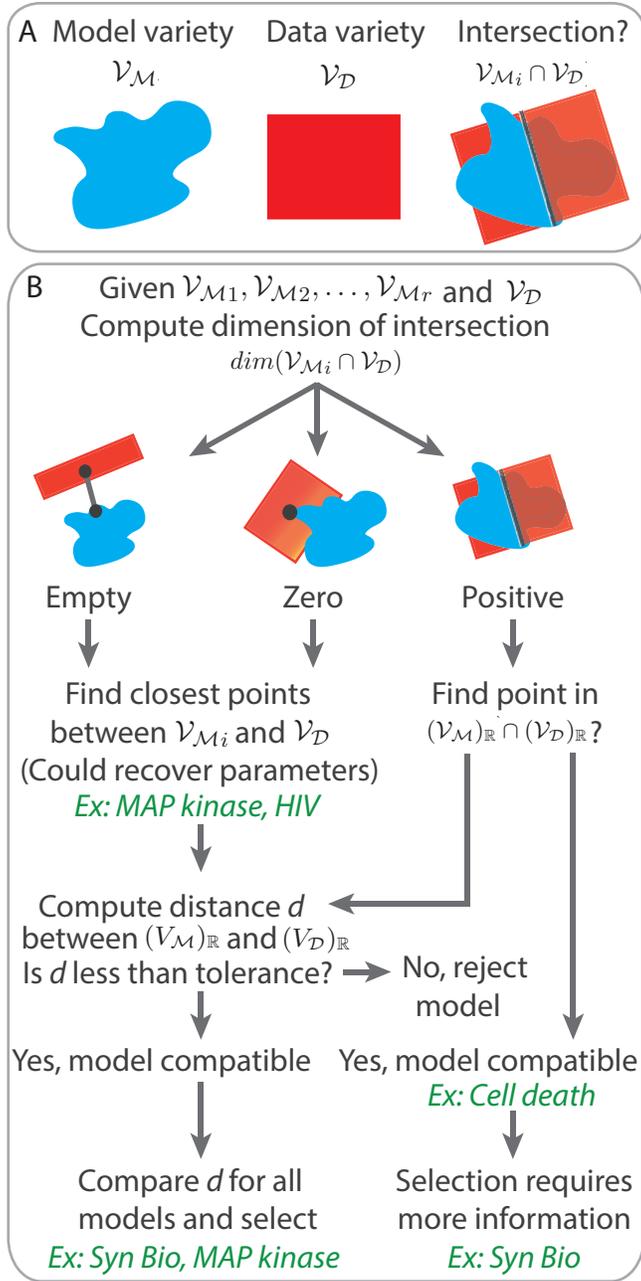


FIGURE 5.1. Schematic of NAG framework corresponding to model validation, selection, and parameter estimation. (A) Input to algorithms include model translated into a model variety (red), and steady-state data translated into a data variety (blue). (B) Flow chart of model compatibility, parameter estimation, and model selection methods. Examples (green) are described in §5.5

5.4.1. **ALGORITHM: MODEL VALIDATION.** The goal of algorithm 2 is to solve the model validation problem posed in §5.2.1. The goal is to find a pair of points that minimize the distance between $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. In the case that $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} = \emptyset$ this is obtained by solving (37)–(39); otherwise a point is selected on real connected components of the nonempty intersection $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$.

Algorithm 2 Model validation

Input: model \mathcal{M} , data $\mathcal{D} = \{\mathbf{y}\}$, tolerance α

Output: *yes* or *no*

- 1: If $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}} = \emptyset$ go to step 3.
 - 2: If $\dim(\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}) \geq 0$ and $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \neq \emptyset$ return *yes*; else go to step 3.
 - 3: Find a pair $((\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}}), (\hat{\mathbf{a}}, \hat{\mathbf{x}}, \mathbf{y})) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \times (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ that minimizes (36) using NAG.
 - 4: If $\|\hat{\mathbf{z}} - \mathbf{y}\|^2 < p_{\alpha}$ return *yes*; else *no*.
-

Determining the dimension of the intersection $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ in steps 1 and 2 of algorithm 2 may be computed several ways. First one could compute $\dim(\mathcal{V}_{\mathcal{M}})$ and $\dim(\mathcal{V}_{\mathcal{D}})$ using the NID from NAG. If $\dim(\mathcal{V}_{\mathcal{M}}) + \dim(\mathcal{V}_{\mathcal{D}})$ exceeds the ambient dimension then they will almost always intersect unless the varieties are in very special positions to one another. For example, two complex 2-planes in \mathbb{C}^3 will almost always intersect unless they are parallel translations to one another. If the ambient dimension exceeds $\dim(\mathcal{V}_{\mathcal{M}}) + \dim(\mathcal{V}_{\mathcal{D}})$ then $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ will often be empty.

A more direct approach is to compute the intersection dimension by solving a corresponding polynomial system of equations. Given data \mathbf{y} replace equation (38) with $\mathbf{y} - \mathbf{g}(\mathbf{x}) = \mathbf{0}$ and solve this together with equation (37). Here what is meant by “solve” is to compute a NID of the corresponding polynomial system of equations using NAG.

In step 2 of algorithm 2 one may be confronted with the possibility that $\dim(\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}) = 0$. In this scenario the intersection of the two varieties consist of finitely-many complex points. The condition $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \neq \emptyset$ indicates that at least one of the points is real which is

straightforward to determine. If $\dim(\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}) > 0$, to check if $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \neq \emptyset$, one needs to apply the technique explained in [37]. The approach of [37] returns a point in the intersection of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ if it exists. If the point has the additional smoothness property then the real dimension of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ is equal to the complex dimension $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$. Smoothness is discussed in §2.1.2 and this technique will be illustrated in §5.5.

To find the pair $((\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}}), (\hat{\mathbf{a}}, \hat{\mathbf{x}}, \mathbf{y}))$ in step 3 of algorithm 2 one solves the polynomial system of equations (37)–(39). If there is a positive-dimensional set of complex critical points then the approach of [37] may return a real point. For example, if a complex plane and line are parallel to one another in \mathbb{C}^3 but do not intersect there is an infinite number of points that minimize the distance; in fact every real point contained on the complex line. The issue of nonnegativity of variables and parameters is discussed in §5.4.2 below.

5.4.2. NONNEGATIVITY CONSIDERATIONS. A common constraint placed on the variables and parameter is that they must be nonnegative. In this case the objective function (36) is minimized over the nonnegative orthant intersected with $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$. Algorithm 2 must be modified in this scenario. If $\dim(\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}) = 0$ or $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}} = \emptyset$ then instead of minimizing the distance between to real algebraic set we minimize the distance between two *semi-algebraic sets* (i.e. sets defined by polynomial equalities and inequalities).

Let $S_{\mathcal{M}} \subset (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ denote the semi-algebraic set associated to the model (i.e. $S_{\mathcal{M}} = \mathcal{V}_{\mathcal{M}} \cap \mathbb{R}_{\geq 0}^{k+n+m}$) then the appropriate statistic is:

$$(40) \quad d^2 = \min \sum_{i=1}^m (z_i - y_i)^2 \text{ subject to } (\mathbf{a}, \mathbf{x}, \mathbf{z}) \in S_{\mathcal{M}}.$$

If only an upper bound on d^2 is sufficient then one could use the test statistic defined in proposition 5.3.1. This would find a local critical point of (36) defined on the interior of $S_{\mathcal{M}}$ but not necessarily its boundary along $\mathbb{R}_{\geq 0}^{k+n+m}$.

If the exact value of d^2 is needed on $S_{\mathcal{M}}$ then one should solve the FJ system of equations.

Let $F_1, \dots, F_r, h_1, \dots, h_s$ be polynomials in the ring:

$$\mathbb{R}[a_1, \dots, a_k, x_1, \dots, x_n, z_1, \dots, z_m].$$

Let $S_{\mathcal{M}}$ be the semi-algebraic set of all $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathbb{R}^{k+n+m}$ that satisfies:

$$F_i(\mathbf{a}, \mathbf{x}, \mathbf{z}) = 0 \text{ for } i = 1, \dots, r,$$

$$h_i(\mathbf{a}, \mathbf{x}, \mathbf{z}) \leq 0 \text{ for } i = 1, \dots, s,$$

and define $\lambda_0, \lambda_1, \dots, \lambda_r, \mu_1, \dots, \mu_s$ as indeterminates corresponding to F_i and h_i , respectively. These indeterminates are called the *FJ multipliers*. The FJ system is written as:

$$(41) \quad \mathbf{F} = \mathbf{0}$$

$$(42) \quad \lambda_0 \begin{bmatrix} \mathbf{0} \\ \mathbf{z} - \mathbf{y} \end{bmatrix} + \sum_{i=1}^r \lambda_i \nabla_{\mathbf{a}, \mathbf{x}, \mathbf{z}} F_i + \sum_{i=1}^s \mu_i \nabla_{\mathbf{a}, \mathbf{x}, \mathbf{z}} h_i = \mathbf{0}$$

$$(43) \quad \mu_1 h_1 = 0$$

$$(44) \quad \vdots$$

$$(45) \quad \mu_s h_s = 0.$$

If $(\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*)$ is a critical point the FJ constraint qualification states that there exists a *nonzero* vector $[\lambda_0, \dots, \lambda_r, \mu_1, \dots, \mu_s]$ with $\mu_i \geq 0$ for $i = 1, \dots, s$ so that:

$$((\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*), [\lambda_0, \dots, \lambda_r, \mu_1, \dots, \mu_s])$$

satisfies equations (41)–(45). To find a global minimum of (40) one uses NAG to solve equations (41)–(45) and then filters solutions appropriately so they satisfy the FJ constraint qualification and constraints. This technique was first employed in [66].

The approach using the FJ system is appropriate to minimize the objective when \mathbf{F} is a complete intersection. However this is not always the case. In the latter one needs to design

an alternative approach in order to employ NAG techniques. The idea is to minimize the objective function over $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ and then check the boundaries of $\mathcal{S}_{\mathcal{M}}$. In other words, remove the explicit inequality constraints $h_i(\mathbf{a}, \mathbf{x}, \mathbf{z}) \leq 0$ from the FJ equations and then minimize the objective along the boundary conditions.

Following proposition 5.3.1 assume that $\mathbf{h}(\mathbf{a}, \mathbf{x}, \mathbf{y})$ are equations that define a complete intersection whose solutions contain $\mathcal{V}_{\mathcal{M}}$ using a complex randomization approach as discussed in §2.1.7 if necessary. First solve the equality constrained optimization problem on $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ using the polynomial system in proposition 5.3.1. Computing d^2 along $S_{\mathcal{M}}$ this way provides an upper bound on (40). The space $\mathbb{R}_{\geq 0}^{k+n+m}$ is naturally a convex polytope made up of faces of various dimensions. Each of the j faces in dimension i , $\mathcal{F}_{i,j}$, is contained in its affine hull $\overline{\mathcal{F}_{i,j}}$, the smallest affine space that contains that face. Over the nonnegative orthant $\overline{\mathcal{F}_{i,j}}$ is defined simply by imposing natural equality constraints. Then one minimizes d^2 over $\mathcal{V}_{\mathcal{M}} \cap \overline{\mathcal{F}_{i,j}}$ for each combination of i, j and filters out solutions not contained in $\mathcal{V}_{\mathcal{M}} \cap \mathbb{R}_{\geq 0}^{k+n+m}$. This is equivalent to minimizing d^2 over $S_{\mathcal{M}}$ once every face of the convex polytope is considered.

If there are N indeterminates there are $2^N - 1$ faces to consider. This amount to solving $2^N - 1$ FJ systems. The number of faces to consider grows very large as $N \rightarrow \infty$. However the dimension of $\mathcal{V}_{\mathcal{M}} \cap \overline{\mathcal{F}_{i,j}}$ is less than or equal to the dimension of $\mathcal{V}_{\mathcal{M}}$ with the inequality being strict when $\mathcal{V}_{\mathcal{M}} \subsetneq \overline{\mathcal{F}_{i,j}}$. If $\mathcal{V}_{\mathcal{M}} \cap \overline{\mathcal{F}_{i,j}}$ is empty then $\mathcal{V}_{\mathcal{M}} \cap \mathcal{S}$ is also empty for any subset $S \subset \overline{\mathcal{F}_{i,j}}$. Using this fact the number of lower-dimensional faces to check is significantly reduced as they are intersections of higher-dimensional faces.

5.4.3. ALGORITHM: MODEL SELECTION. During model selection there are several competing models whos steady-state equilibria are defined by distinct polynomial systems. Algorithm 2 is applied for each model under consideration. If a significance level α is prescribed

then first reject any model that does not support the observed data ($d^2 \geq p_\alpha$) and then a model is selected that minimizes the value d^2 .

5.4.4. ALGORITHM: PARAMETER ESTIMATION. The algorithm for parameter estimation is built upon algorithm 2. In this situation only the model \mathcal{M} and data \mathcal{D} are prescribed. One assumes that there are unknown parameters to be estimated and parameters are produced that are the best fit between the model \mathcal{M} and the data \mathcal{D} . The outputs for step 4 of algorithm 2 is removed from the algorithm as there is no significant level α being considered in parameter estimation. Instead of returning *yes* or *no* algorithm 2 is modified to return simply the value $(\hat{\mathbf{a}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})$ for which the estimated parameters $\hat{\mathbf{a}}$ may be recovered.

5.4.5. ILLUSTRATIVE EXAMPLE. Before looking at more complex examples in §5.5 we first consider a simple model to illustrate algorithm 2. Consider a model with three variables x, y, z and three parameters a, b, c satisfying the equation for an ellipse centered at the origin:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1.$$

Consider first a significance level of $\alpha = 0.1$ and variances on the errors $\sigma_i^2 = 0.1$. The model variety $\mathcal{V}_{\mathcal{M}}$ is illustrated in figure 5.2(A). Now set $a, b, c = 1$.

Suppose the outputs are x, y , and z is an unobserved variable. We make an observation of $x', y' = 0$. One concludes that $\mathcal{V}_{\mathcal{D}}$ is the z -axis and it intersects $\mathcal{V}_{\mathcal{M}}$ at two points illustrated in figure 5.2(B). Following along with algorithm 2 step 2 tell us *yes* the model is compatible with the data.

Now suppose instead an observation of $x' = 0$ is made and y, z are unobserved variables. In this case $\mathcal{V}_{\mathcal{D}}$ becomes the yz -axis. Due to the simple geometry we know the intersection contains real points so the model would be compatible from step 2 of algorithm 2. For the sake of argument the NID of the intersection $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ would have dimension one. We

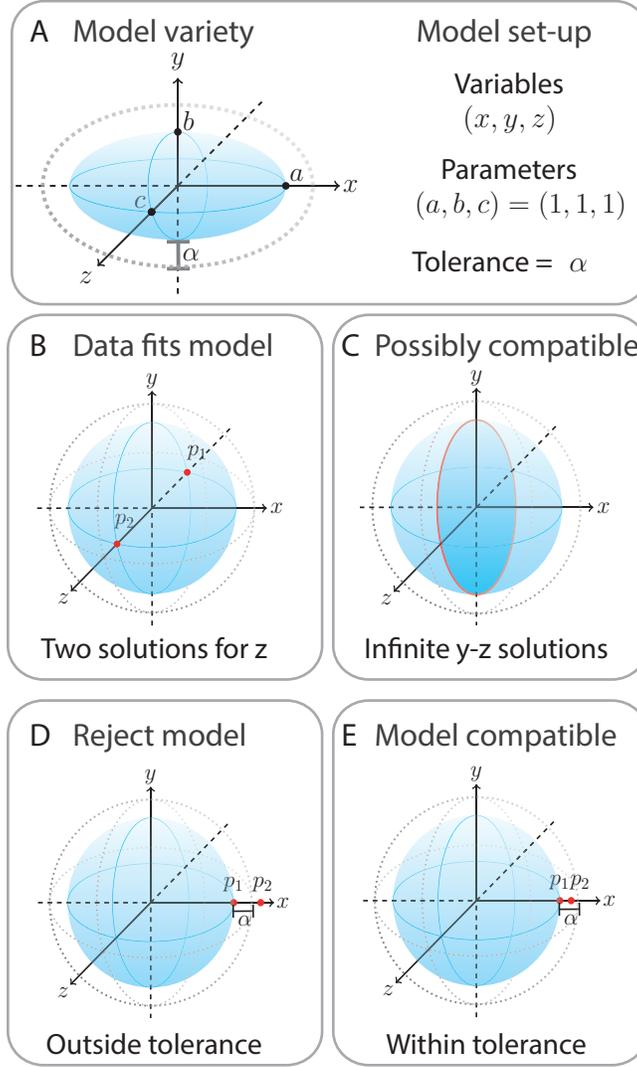


FIGURE 5.2. Simple example demonstrating model compatibility following algorithm 1.

could then determine compatibility using the approach of [37] by finding real points on each connected component of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. This is illustrated in figure 5.2(C).

Now suppose an observation of $x' = 1.7$ and $y' = 0$ is made so that z is an unobserved variable. The model variety defines a line parallel to the z -axis and does not intersect $\mathcal{V}_{\mathcal{M}}$ over \mathbb{R} . Therefore one applies step 3 of algorithm 2 and finds a point $((1, 0, 0), (1.7, 0, 0)) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \times (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. Thus the scaled test statistic is:

$$d^2 = \frac{(x - x')^2}{\sigma_1^2} + \frac{(y - y')^2}{\sigma_2^2} = \frac{(1 - 1.7)^2}{0.1} + \frac{(0 - 0)^2}{0.1} = 4.9.$$

The upper 0.1 percentile of χ^2 with two degrees of freedom is $p_{0.1} = 0.4605$. Therefore since $d^2 > p_{0.1}$ we reject the null hypothesis that the model is compatible with the data. This is illustrated in figure 5.2(D).

Similarly if $x' = 1.01$ and $y' = 0$ then again $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ is empty and step 3 of algorithm 2 is applied and a point $((1, 0, 0), (1.01, 0, 0)) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \times (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ is found. In this case $d^2 = 0.001$ so that $d^2 < p_{0.1} = 0.4605$. We fail to reject the null hypothesis and conclude that the model is compatible with the data. This is illustrated in figure 5.2(E).

5.5. RESULTS AND EXPERIMENTS

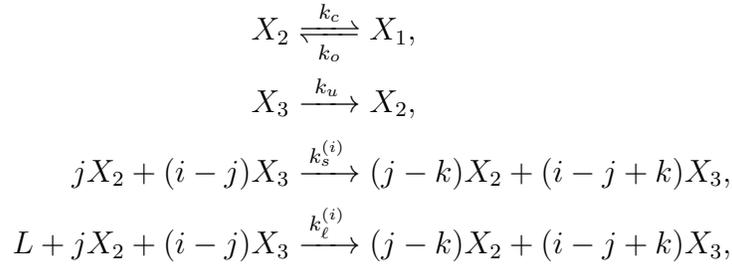
The NAG approach to model selection will be demonstrated in three examples: cell death activation, human immunodeficiency virus (HIV) progression, and multisite phosphorylation. Each model is governed by a 1st order polynomial ODE of the form $\mathbf{x}'(t) = \mathbf{f}(\mathbf{a}, \mathbf{x})$ at steady state. The aim of this section is to explain the core computational steps required in applying model validation, selection, or parameter estimation. Where appropriate an explanation of the findings will also be explained.

In §5.5.1 we look at a model of receptor-mediated programmed cell death and illustrate how model compatibility can be determined from simulated data when the dimension of intersection of the model and data variety is positive. Then in §5.5.2 we look at a model describing long-term HIV dynamics and show how the natural death of HIV parameter may be estimated from simulated data. Finally in §5.5.3 we study two conflicting models of phosphorylation mechanisms of cellular signaling. Using experimental data from *in vitro* and *in vivo* measurements we illustrate model selection and parameter estimation.

5.5.1. CELL DEATH ACTIVATION. Consider a model of receptor-mediated programmed cell death initiated by activation of death receptors under detection of extracellular death

ligands via the Fas mechanism [27, 57, 73]. There are a variety of models to explain the data and in this example we choose to study the so-called *cluster* model [42]. The cluster model is interesting in that it exhibits the phenomena of bistability in equilibria. The aim of this example is to illustrate model validation in the scenario that the intersection of the model and data varieties are nonempty.

Setting up the model. The model includes a variety of mechanisms including constitutive receptor opening and closing, pairwise open Fas stabilization, higher-order open Fas stabilization enabled by FasL, and ligand-induced receptor opening. Fas is assumed to be one of three species: closed (X_1); open, unstable (X_2); and open, stable (X_3). Suppose the ligand FasL is denoted by the variable L . The cluster model is defined using the reactions:



for $i \in \{2, 3\}$, $j = 1, \dots, i$, and $k = 1, \dots, j$. The first reaction defines receptor opening and closing. The second reaction describes destabilization of open Fas. The third reaction(s) define higher-order cluster-stabilization by open Fas independent of the ligand FasL and similarly for the fourth reaction(s) dependent on the ligand FasL.

Assuming the reactions behave according to mass-action kinetics we may translate to a system of 1st order ODEs:

$$(46) \quad x'_1(t) = -v_1,$$

$$(47) \quad x'_2(t) = v_1 + v_2 - v_3 - v_4,$$

$$(48) \quad x'_3(t) = v_3 + v_4 - v_2,$$

where:

$$\begin{cases} v_1 &= k_o x_1 - k_c x_2, \\ v_2 &= k_u x_3, \\ v_3 &= 6k_s^{(3)} x_2^3 + 3k_s^{(3)} x_2^2 x_3 + 3k_s^{(2)} x_2^2 + k_s^{(3)} x_2 x_3^2 + k_s^{(2)} x_2 x_3, \\ v_4 &= 6k_\ell^{(3)} x_2^3 \ell + 3k_\ell^{(3)} x_2^2 x_3 \ell + 3k_\ell^{(2)} x_2^2 \ell + k_\ell^{(3)} x_2 x_3^2 \ell + k_\ell^{(2)} x_2 x_3 \ell. \end{cases}$$

Here v_i define the reaction velocities for the variables x_i . Lowercase letters denote the concentrations of their corresponding species defined in the model reactions above. The model parameters for the cell death cluster model are:

$$\mathbf{a} = (k_o, k_c, k_u, k_s^{(2)}, k_s^{(3)}, k_\ell^{(2)}, k_\ell^{(3)}),$$

the variables are:

$$\mathbf{x} = (\ell, x_1, x_2, x_3),$$

and the outputs are:

$$\mathbf{z} = (\lambda, \rho, \zeta).$$

The outputs represent the total ligand concentration, total receptor concentration, and the total downstream “death signal”, respectively, as given by the equations:

$$(49) \quad \lambda - \ell = 0,$$

$$(50) \quad \rho - (x_1 + x_2 + x_3) = 0,$$

$$(51) \quad \zeta - x_3 = 0.$$

The model variety \mathcal{V}_M may be constructed as the zero set of equations defined by setting the right hand side of (46)–(48) to zero together with equations (49)–(51). A simple dimension count shows that \mathcal{V}_M is contained in \mathbb{C}^{14} whose coordinates are defined by the model parameters \mathbf{a} , variables \mathbf{x} , and outputs \mathbf{z} .

Given an observable data point $\mathbf{y} = (\lambda', \rho', \zeta')$ the data variety is defined as:

$$(52) \quad \mathcal{V}_{\mathcal{D}} = \{(\mathbf{x}, \mathbf{a}, \mathbf{z}) \in \mathbb{C}^{14} : \lambda = \lambda', \rho = \rho', \zeta = \zeta'\}.$$

$\mathcal{V}_{\mathcal{D}}$ has dimension 11 since there are zero degrees of freedom in the variables $\lambda, \rho,$ and ζ .

Returning to $\mathcal{V}_{\mathcal{M}}$ a simple codimension count provides a lower bound on the dimension of $\mathcal{V}_{\mathcal{M}}$. We expect the dimension of $\mathcal{V}_{\mathcal{M}}$ to be at least $14 - 6 = 8$ assuming the equations are consistent (i.e. the ideal generated by the polynomials are not equal to the ideal (1)). In order to be more precise one could compute a NID of $\mathcal{V}_{\mathcal{M}}$. Using Bertini [8] shows that $\mathcal{V}_{\mathcal{M}}$ is a 9-dimensional complex set of degree 10.

A steady-state data point:

$$\mathbf{y} = (\lambda', \rho', \zeta') = (1.7784308, 2.31883024, 2.16896112)$$

was simulated from the ODE model (46)–(48) with all parameters and initial concentrations taken as i.i.d. draws from the log-normal distribution $\ln \mathcal{N}(0, 4)$ then combined and corrupted with i.i.d. noise from $\mathcal{N}(0, 0.1)$ to obtain \mathbf{y} .

As discussed in §5.4.1 since $\dim \mathcal{V}_{\mathcal{M}} + \dim \mathcal{V}_{\mathcal{D}} = 9 + 11 > 14 = \dim \mathbb{C}^{14}$ we expect $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}} \neq \emptyset$. A direct computation of the NID shows that $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ is a 6-dimensional complex algebraic set of degree 5. Modifying the observable data point \mathbf{y} by adding noise to each coordinate drawn from the distribution $\mathcal{N}(0, 0.1)$ did not affect the dimension or degree.

At this point step 1 of algorithm 2 is complete and what has been shown is that $\dim(\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}) > 0$. The intersection provides evidence that $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \neq \emptyset$ but there is no guarantee. Our goal moving forward is to find at least one nonnegative point in $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$.

Fritz John Conditions. In order to find real points we apply the methods described in [37]. First randomly select a real positive point $(\mathbf{a}^*, \mathbf{x}^*)$ whose coordinates are chosen on a nonnegative closed interval. The point will determine \mathbf{z}^* using equations (49)–(51). The aim is then to solve the constrained optimization problem:

$$(53) \quad \begin{aligned} & \text{minimize} && \|(\mathbf{a}, \mathbf{x}, \mathbf{z}) - (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*)\|^2 \\ & \text{subject to} && (\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}. \end{aligned}$$

Geometrically optimization problem (53) is minimizing the distance between $(\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*)$ and $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$.

In what follows the aim is to set up the polynomial system of equations that finds all critical points of (53). We take a perturbed regeneration approach to solving the defining system. Perturbed regeneration is discussed in chapter 3. Refer to the system defining $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ as $\mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z})$.

The first subroutine is to “square up” \mathbf{f}^* . The theoretical foundation of squaring up a system is discussed in §2.1.7. First we compute the codimension of $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ using the information gained above and observe the codimension is $14 - 6 = 8$. There exists a nonempty Zariski open set $\mathcal{A} \subseteq \mathbb{C}^{8 \times 9}$ such that for every matrix $\mathbf{A} \in \mathcal{A}$, we have $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}} \subseteq \mathcal{V}(\mathbf{A}\mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z}))$. Elements of matrices \mathbf{A} are chosen uniformly along the complex unit circle. If a point $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}(\mathbf{A}\mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z}))$ then $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ may be verified by function evaluation of \mathbf{f}^* . Furthermore, if $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ contains a smooth real point then $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ has real dimension six.

The polynomial system to find all critical points of (53) is the so-called FJ conditions for optimality. That is, if $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ is a critical point of (53) it must satisfy the FJ conditions. We would like a homotopy that is designed to find all solutions that satisfy these conditions.

Setting up the Homotopy. Given a point $(\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*) \in \mathbb{R}^{14}/(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ the FJ condition for optimality states that $\boldsymbol{\xi} = (\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ is a local critical point of $\|(\mathbf{a}, \mathbf{x}, \mathbf{z}) - (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*)\|^2$ if there exists a $\tilde{\boldsymbol{\lambda}} \in \mathbb{P}^8$, complex projective space, so that $(\boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}} \times \mathbb{P}^8$ satisfies:

$$(54) \quad \mathbf{A}\mathbf{f}^* = \mathbf{0},$$

$$(55) \quad ((\mathbf{a}, \mathbf{x}, \mathbf{y}) - (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*))^T \lambda_0 + J(\mathbf{A}\mathbf{f}^*)^T(\lambda_1, \dots, \lambda_8)^T = \mathbf{0},$$

where $\boldsymbol{\lambda} = [\lambda_0, \dots, \lambda_8] \in \mathbb{P}^8$ and $J(\mathbf{A}\mathbf{f}^*)$ denotes the Jacobian matrix of the functions $\mathbf{A}\mathbf{f}^*$ with respect to the variables $(\mathbf{a}, \mathbf{x}, \mathbf{z})$. A generic affine patch of \mathbb{P}^8 is then chosen so that the system may be solved using affine coordinates. This is a necessary step in order to perform homotopy continuation. More specifically there is a nonempty Zariski open subset $\mathcal{B} \in \mathbb{C}^9$ so that for every $\boldsymbol{\alpha} \in \mathcal{B}$ the FJ conditions may be solved in affine coordinates:

$$(56) \quad \mathbf{A}\mathbf{f}^* = \mathbf{0},$$

$$(57) \quad ((\mathbf{a}, \mathbf{x}, \mathbf{z}) - (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*))^T \lambda_0 + J(\mathbf{A}\mathbf{f}^*)^T(\lambda_1, \dots, \lambda_8)^T = \mathbf{0},$$

$$(58) \quad \alpha_0 \lambda_0 + \alpha_1 \lambda_1 + \dots + \alpha_8 \lambda_8 - 1 = 0.$$

The components of $\boldsymbol{\alpha}$ are chosen uniformly on the complex unit circle. Since $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$ is a complex six-dimensional algebraic set using witness sets the hypotheses of Theorem 5 [37] apply.

The results of theorem 5 [37] are as follows. Let $\mathbf{w} \in \mathbb{R}^8, \gamma \in \mathbb{C}$, and homotopy $\mathbf{H} : \mathbb{C}^{14} \times \mathbb{C}^9 \times \mathbb{C} \rightarrow \mathbb{C}^{23}$ be defined by:

$$(59) \quad \mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, t) = \begin{bmatrix} \mathbf{A}\mathbf{f}^* - t\gamma\mathbf{w} \\ ((\mathbf{a}, \mathbf{x}, \mathbf{z}) - (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*))^T \lambda_0 + J(\mathbf{A}\mathbf{f}^*)^T(\lambda_1, \dots, \lambda_8)^T \\ \alpha_0 \lambda_0 + \alpha_1 \lambda_1 + \dots + \alpha_8 \lambda_8 - 1 \end{bmatrix}.$$

\mathbf{H} therefore has the properties that the roots of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, 1)$ are finite and nonsingular, the number of solutions of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, 1) = \mathbf{0}$ is maximal for generically-chosen

$\mathbf{w}, \gamma, (\mathbf{a}^*, \mathbf{x}^*, \mathbf{z}^*)$, and $\boldsymbol{\alpha}$. Furthermore, the one real-dimensional solution paths defined by the homotopy \mathbf{H} starting at $t = 1$ are capable of computing the solutions $\boldsymbol{\xi}$ that satisfy the FJ condition after projecting solutions $(\boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}})$ onto $\boldsymbol{\xi}$ as $t \rightarrow 0$. The polynomial system (59) consists of 23 variables and equations.

Finding Solutions. The nonsingular isolated solutions of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, 1) = \mathbf{0}$ are computed numerically using regeneration. Regeneration is discussed in detail in §3.3. As explained in §3.3 regeneration is more appropriate when only nonsingular solutions are desired. Since theorem 5 [37] only requires nonsingular solutions at $t = 1$ this approach is appropriate. After the nonsingular roots of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, 1)$ are obtained a straight-line parameter homotopy is performed and critical points that satisfy the FJ conditions are found approximately as $t \rightarrow 0$. Parameter homotopies are discussed in §2.4.

All of the subexpressions of (49)–(51) are affine linear. In this scenario employing the use of intrinsically-defined variables significantly reduces computation time. For example, the regeneration and parameter homotopy routines are run explicitly using a subset of the variables and parameters $x_1, k_o, k_c, k_u, k_s^{(2)}, k_s^{(3)}, k_\ell^{(2)}, k_\ell^{(3)}$. The other variables and outputs, $\ell, x_2, x_3, \lambda, \rho, \zeta$ are parameterized in terms of the others.

Timing and Implementation. A timing summary may be found in table 5.1. Timings include computing the NID of \mathcal{V}_M and $\mathcal{V}_M \cap \mathcal{V}_D$ and the two steps to approximate the critical points. The NIDs and parameter homotopy were implemented on a Apple MacBook

TABLE 5.1. Timings collected over 20 runs. The table includes the average time and standard deviations associated to the four computations described in this section.

	Timing
Compute \mathcal{V}_M	0.79 sec \pm 0.10 sec
Compute $\mathcal{V}_M \cap \mathcal{V}_D$	0.35 sec \pm 0.10 sec
Regeneration (parallel)	13.69 sec \pm 2.40 sec
Parameter homotopy	0.04 sec

Pro with 2.4 GHz Intel “Core i5” processor using a serial implementation of Bertini [8]. Regeneration was implemented on 24 (2.67 GHz Xeon-5650) compute nodes with a CentOS 5.11 OS using a parallel implementation of Bertini [8].

Interpreting the output. After approximating and examining critical points there are three solutions that correspond to real points on $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$. Among the three solutions two are nonnegative. One then verifies these are solutions to $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ by function evaluation of $\mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z})$ and thus are also solutions of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$. It is interesting to speculate if the two positive real solutions found were contained on distinct connected components of $(\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap (\mathcal{V}_{\mathcal{D}})_{\mathbb{R}}$ corresponding to the the bistable “branches”. Algorithm 2 is then completely applied and one concludes that the clustering model $\mathcal{V}_{\mathcal{M}}$ is compatible with the observable data y .

Concerns with the approach. The first concern is that model compatibility may need to be determined for a large set of outputs rather than just one. This issue is easily addressed by employing a parameter homotopy scheme. In this case since regeneration only needs to be applied once every instance of outputs requires simply a straight-line homotopy that takes on the order of 0.03 seconds to complete for the clustering model.

The second concern is that the intersection of the model and data varieties, $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$, may be composed of several complex components of varying dimension. In addition each pure-dimensional component may consist of several irreducible components that are either conjugate pairs or self-conjugate. In the later case real points of $(\mathcal{V}_{\mathcal{M}}) \cap (\mathcal{V}_{\mathcal{D}})$ are contained on the intersection of these components whose real dimension is less than expected. These issues are addressed using the theory from [37] guaranteeing that at least one real point is obtained on each real connected component.

In a systematic way one constructs systems of the form $\mathbf{A}\mathbf{f}^*$ via randomization so that the codimension of $\mathcal{V}(\mathbf{A}\mathbf{f}^*)$ pertains to the dimension of each pure-dimensional component of $\mathcal{V}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{D}}$. When there are several pure-dimensional component a homotopy must be solved for each dimension.

5.5.2. HIV PROGRESSION. In this example we illustrate parameter estimation discussed in §5.4.4. The parameter estimation algorithm is build upon algorithm 2 related to model validation. The model considered aims to model long-term HIV dynamics from initial viremia, latency, and virus increase [41, 33].

In this model the HIV virus inhibits the CD4+T cell population while promoting macrophage proliferation, and eventually houses the replicating virus. As macrophages proliferate the virus reservoir increases so the model describes a HIV patients progression to AIDS. The model can have two real equilibria [33], one of which is stable, representing patients that are “long-term non-progressors” [41].

Setting up the model. Model variables \mathbf{x} are uninfected CD4+ T cells (T), infected CD4+ T cells (T_i), uninfected macrophages (M), infected macrophages (M_i), and HIV virus population (V). The parameters \mathbf{a} are $(s_1, s_2, k_1, \dots, k_6, \delta_1, \dots, \delta_5)$ where s_i represents synthesis of T cells and macrophages, k are rate constants describing interactions between variables \mathbf{x} , and δ_i represents natural death parameters of the model variables \mathbf{x} , respectively. We assume that all of the variables are measurable outputs so that $\mathbf{y} = \mathbf{x}$.

The reactions for the HIV model are summarized in table 5.2. Assuming mass-action kinetics the reactions from table 5.2 may be translated into a 1st order system of ODEs:

$$(60) \quad T'(t) = s_1 + k_1TV - k_2TV - \delta_1T,$$

$$(61) \quad T'_i(t) = k_2TV - \delta_2T_i,$$

$$(62) \quad M'(t) = s_2 + k_3MV - k_4MV - \delta_3M,$$

$$(63) \quad M'_i(t) = k_4MV - \delta_4M_i,$$

$$(64) \quad V'(t) = k_5T_i + k_6M_i - \delta_5V.$$

The model variety $\mathcal{V}_{\mathcal{M}}$ can be constructed by considering the zero set of the polynomials defined by the right hand side of equations (61)–(64). By computing an NID of $\mathcal{V}_{\mathcal{M}}$ one finds two irreducible components \mathcal{V}_1 and \mathcal{V}_2 . More specifically generators for the ideal of these

TABLE 5.2. Reactions for HIV model. The parameter values used are from [41].

Description	Reaction	Parameter
Generation of new CD4+T cells	$\emptyset \xrightarrow{s_1} T$	10
Generation of new macrophages	$\emptyset \xrightarrow{s_2} M$	0.15
Proliferation of T cells by presence of pathogen	$T+V \xrightarrow{k_1} (T+V)+T$	0.002
Infection of T cells by HIV	$T+V \xrightarrow{k_2} T_i$	0.003
Proliferation of M by presence of pathogen	$M+V \xrightarrow{k_3} (M+V)+M$	0.000745
Infection of M by HIV	$M+V \xrightarrow{k_4} M_i$	0.000522
Proliferation of HIV within CD4+T cell	$T_i \xrightarrow{k_5} V+T_i$	0.537
Proliferation of HIV within macrophage	$M_i \xrightarrow{k_6} V+M_i$	0.285
Natural death of CD4+T cells	$T \xrightarrow{\delta_1} \emptyset$	0.01
Natural death of infected T cells	$T_i \xrightarrow{\delta_2} \emptyset$	0.44
Natural death of macrophages	$M \xrightarrow{\delta_3} \emptyset$	0.0066
Natural death of infected macrophages	$M_i \xrightarrow{\delta_4} \emptyset$	0.0066
Natural death of HIV	$V \xrightarrow{\delta_5} \emptyset$	3

components are:

$$\begin{aligned} \mathcal{I}(\mathcal{V}_1) = \langle & 5742M - 2453M_i - 130500, \\ & 259908T_i - 46607M_i + 4840000\delta_5 - 20200500, \\ & 17721T + 46607M_i - 4840000\delta_5 + 2479500, \\ & 484000V\delta_5 - 184547M_i + 4840000\delta_5 - 20200500, \\ & 2453M_iV - 72600M_i + 130500V \rangle \end{aligned}$$

and

$$\mathcal{I}(\mathcal{V}_2) = \langle V, M_i, 11M-250, T_i, T-1000 \rangle.$$

\mathcal{V}_2 is called an *extinction component* and is not interesting to use for parameter estimation. Instead we replace \mathcal{V}_M with \mathcal{V}_1 and use this to perform parameter estimation.

Estimating natural death of HIV. Using the model variety \mathcal{V}_1 our aim is to estimate the natural death of HIV parameter δ_5 . We use the long-term nonprogression steady-state values from table 3 of [41] to construct \mathbf{y} and corrupt these values with noise drawn from $\mathcal{N}(0, 1)$. In particular the data variety, \mathcal{V}_D , is defined using $\mathbf{y} = (\frac{6383}{20}, \frac{937}{20}, \frac{8109}{100}, \frac{13667}{100}, \frac{2121}{100})$. Furthermore $s_1, s_2, k_1, \dots, k_6, \delta_1, \dots, \delta_4$ are treated as known parameters using the values from table 1 of [41]. One may verify that $\mathcal{V}_1 \cap \mathcal{V}_D = \emptyset$ by computing a NID or showing that the ideal $\mathcal{I}(\mathcal{V}_1 \cap \mathcal{V}_D) = (1)$ using a computer algebra system such as Macaulay2 [30]. Using Bertini [8] one may then solve the system from proposition 5.3.1:

$$(65) \quad 5742M - 2453M_i - 130500 = 0,$$

$$(66) \quad 259908T_i - 46607M_i + 4840000\delta_5 - 20200500 = 0,$$

$$(67) \quad 17721T + 46607M_i - 4840000\delta_5 + 2479500 = 0,$$

$$(68) \quad 484000V\delta_5 - 184547M_i + 4840000\delta_5 - 20200500 = 0,$$

$$(69) \quad 2453M_iV - 72600M_i + 130500V = 0,$$

$$(70) \quad T + 17721\lambda_3 - 6383/20 = 0,$$

$$(71) \quad T_i + 259908\lambda_2 - 937/20 = 0,$$

$$(72) \quad M + 5742\lambda_1 - 8109/100 = 0,$$

$$(73) \quad 2453\lambda_5 + M_i - 2453\lambda_1 - 46607\lambda_2 + 46607\lambda_3 - 184547\lambda_4 - 72600\lambda_5 - 13667/100 = 0,$$

$$(74) \quad 484000\delta_5\lambda_4 + 2453M_i\lambda_5 + V + 130500\lambda_5 - 2121/100 = 0,$$

$$(75) \quad 484000V\lambda_4 + 4840000\lambda_2 - 4840000\lambda_3 + 4840000\lambda_4 = 0.$$

Solving equations (66)–(75) produces 16 complex solutions where three of which are real solutions. The real solution that minimizes the test statistic d^2 estimates a value of the natural death of HIV to be $\bar{\delta}_5 \approx 2.99876$ which is approximately the same as the true value $\delta_5 = 3$. Computations took 48 seconds total and were performed on an Apple MacBook Pro with a 2.6 GHz Intel Core i5 processor.

5.5.3. MULTISITE PHOSPHORYLATION. In this example we look into the phosphorylation mechanisms of cellular signaling using experimental data. The goal of this example is to illustrate how the model selection and parameter estimation algorithms are applied. Furthermore we will analyze the results of the output of the model selection algorithms.

Phosphorylation is a key cellular regulatory mechanism that has been studied both experimentally and theoretically [28]. One aspect of interest in the mechanism by which a kinase phosphorylates a two-site substrate. For example, the kinase could phosphorylate *distributively* where the kinase adds at most one phosphate before dissociating. However, the kinase could also phosphorylate *processively* where it can add both phosphates in sequence.

The so-called MAPK/ERK pathway is a well-known system for studying phosphorylation where MEK (kinase) phosphorylates ERK (the substrate). There is experimental evidence using polynomial ODEs that suggests that the mammalian MAPK/ERK pathway acts distributively *in vitro* but acts processively *in vivo* [4].

Setting up the experiment. We will consider 12 different levels of EGF stimulus ranging from 0.0244140625 ng/mL to 50 ng/mL. We study EGF stimulus because EGF activates cRAF which then phosphorylates MEK and finally doubly phosphorylates ERK. The observable data consists of measurements of three replicates of: nonphosphorylated ERK (np-ERK), tyrosine monophosphorylated ERK (pY-ERK), and doubly phosphorylated ERK (pTpY-ERK) at each stimulus level. Data is given as a percentage of total ERK (ERK_{tot})

and one uses the concentration measurement for each of these ERK states. The goal is to understand what model, either processive or distributive, best explains EGF stimulus using the approach explain in §5.4.2 and in addition perform parameter estimation.

Mathematical models. Model variables are given in table 5.3, and model parameters are given in table 5.4. The model parameters for the distributive model are:

$$\mathbf{a} = (k_1, \dots, k_{27}, c_1, c_2),$$

the variables are:

$$\mathbf{x} = (x_1, \dots, x_{12}, \text{cRAF}_{tot}, \text{MEK}_{tot}, \text{ERK}_{tot}),$$

and the outputs are:

$$\mathbf{z} = (\text{np-ERK}, \text{pY-ERK}, \text{pYpT-ERK}).$$

The variables for the processive model are the same as for the distributive model except

TABLE 5.3. Description of variables for distributive and processive MAP Kinase models.

variable	species	variable	species
x_1	MEK	x_8	pY-ERK_nuc
x_2	cRAF	x_9	pT-ERK_cyt
x_3	pMEK	x_{10}	pT-ERK_nuc
x_4	np-ERK_cyt	x_{11}	pTpY-ERK_cyt
x_5	MEK_np-ERK	x_{12}	pTpY-ERK_nuc
x_6	np-ERK_nuc	x_{13}	pMEK_np-ERK
x_7	pY-ERK_cyt	x_{14}	pMEK_pY-ERK

for two additional variables x_{13}, x_{14} . The reaction velocities are given in table 5.5 and the corresponding equations are given in table 5.6. *In vitro* parameters estimates are used from table S2 of [4] for $k_2, \dots, k_{27}, c_1, c_2$ and the conserved quantities $\text{MEK}_{tot}, \text{cRAF}_{tot}, \text{ERK}_{tot}$ are listed in table 5.7. The unknown parameter k_1 describes the rate of MEK phosphorylation and depends on the level of EGF stimulation which varies from the output data.

TABLE 5.4. Description of parameters for distributive and processive MAP Kinase models.

parameter	name	parameter	name
k_1	kphos_MEK_pMEK	k_{15}	kdphos_pY_np_cyt
k_2	kdphos_pMEK_MEK	k_{16}	kdphos_pT_np_cyt
k_3	kf_MEK_ERK_binding	k_{17}	kdphos_pTpY_pY_nuc
k_4	kb_MEK_ERK_dissociation	k_{18}	kdphos_pTpY_pT_nuc
k_5	kimport_np	k_{19}	kdphos_pY_np_nuc
k_6	kexport_np	k_{20}	kdphos_pT_np_nuc
k_7	kimport_pY	k_{21}	kphos_np_pY
k_8	kexport_pY	k_{22}	kphos_pY_pTpY
k_9	kimport_pT	k_{23}	kphos_pT_pTpY
k_{10}	kexport_pT	k_{24}	kf_MEK_ERK_binding
k_{11}	kimport_pTpY	k_{25}	kb_MEK_ERK_dissociation
k_{12}	kexport_pTpY	k_{26}	kphos_np_pY
k_{13}	kdphos_pTpY_pY_cyt	k_{27}	kphos_pY_pTpY_MEKERK
k_{14}	kdphos_pTpY_pT_cyt	c_2, c_1	cyt_vol, nuc_vol

TABLE 5.5. Reaction velocities for the MAP Kinase distributive and processive model. The processive model uses the additional reaction velocities v_{18}, v_{19}, v_{20} .

$v_1 = k_1 x_1 x_2 - k_2 x_3$	$v_2 = k_3 x_1 x_4 - k_4 x_5$	$v_3 = k_5 x_4 - c_2 k_6 x_6$
$v_4 = k_7 x_7 - c_2 k_8 x_8$	$v_5 = k_9 x_9 - c_2 k_{10} x_{10}$	$v_6 = k_{11} x_{11} - c_2 k_{12} x_{12}$
$v_7 = k_{13} x_{11}$	$v_8 = k_{14} x_{11}$	$v_9 = k_{15} x_7$
$v_{10} = k_{16} x_9$	$v_{11} = c_2 k_{17} x_{12}$	$v_{12} = c_2 k_{18} x_{12}$
$v_{13} = c_2 k_{19} x_8$	$v_{14} = c_2 k_{20} x_{10}$	$v_{15} = k_{21} x_3 x_4$
$v_{16} = k_{22} x_3 x_7$	$v_{17} = k_{23} x_3 x_9$	
$v_{18} = k_{24} x_3 x_4 - k_{25} x_{13}$	$v_{19} = k_{26} x_{13}$	$v_{20} = k_{27} x_{14}$

The output variables are np-ERK, pY-ERK, and pYpT-ERK which are sums of species concentrations. For the distributive model the output equations are:

$$(76) \quad \text{np-ERK} - (x_4 + x_5 + x_6) = 0,$$

$$(77) \quad \text{pY-ERK} - (x_7 + x_8) = 0,$$

$$(78) \quad \text{pYpT-ERK} - (x_{11} + x_{12}) = 0,$$

whereas for the processive model we include two additional species:

$$(79) \quad \text{np-ERK} - (x_4 + x_5 + x_6 + x_{13}) = 0,$$

$$(80) \quad \text{pY-ERK} - (x_7 + x_8 + x_{14}) = 0,$$

$$(81) \quad \text{pYpT-ERK} - (x_{11} + x_{12}) = 0.$$

TABLE 5.6. Equations for distributive and processive MAP Kinase models.

Variable	Distributive	Processive
$x'_1 =$	$-v_1 - v_2$	$-v_1 - v_2$
$x'_2 =$	0	0
$x'_3 =$	v_1	$v_1 - v_{18} + v_{20}$
$x'_4 =$	$-v_2 - v_3 + v_9 + v_{10} - v_{15}$	$-v_2 - v_3 + v_9 + v_{10} - v_{18}$
$x'_5 =$	v_2	v_2
$x'_6 =$	$v_3 + v_{13} + v_{14}$	$v_3 + v_{13} + v_{14}$
$x'_7 =$	$-v_4 + v_7 - v_9 + v_{15} - v_{16}$	$-v_4 + v_7 - v_9 - v_{16}$
$x'_8 =$	$v_4 + v_{11} - v_{13}$	$v_4 + v_{11} - v_{13}$
$x'_9 =$	$-v_5 + v_8 - v_{10} - v_{17}$	$-v_5 + v_8 - v_{10} - v_{17}$
$x'_{10} =$	$v_5 + v_{12} - v_{14}$	$v_5 + v_{12} - v_{14}$
$x'_{11} =$	$-v_6 - v_7 - v_8 + v_{16} + v_{17}$	$-v_6 - v_7 - v_8 + v_{16} + v_{17} + v_{20}$
$x'_{12} =$	$v_6 - v_{11} - v_{12}$	$v_6 - v_{11} - v_{12}$
$x'_{13} =$		$v_{18} - v_{19}$
$x'_{14} =$		$v_{19} - v_{20}$
$0 =$	$\text{MEK}_{tot} - (x_1 + x_3 + x_5)$	$\text{MEK}_{tot} - (x_1 + x_3 + x_5 + x_{13} + x_{14})$
$0 =$	$\text{cRAF}_{tot} - x_2$	$\text{cRAF}_{tot} - x_2$
$0 =$	$\text{ERK}_{tot} - \sum_{i=4}^{12} x_i$	$\text{ERK}_{tot} - \sum_{i=4}^{14} x_i$

TABLE 5.7. Parameter values for MAP Kinase models

parameter	value	parameter	value	parameter	value
k_2	0.0096	k_{13}	0.004	k_{24}	0.18
k_3	0.18	k_{14}	0.0055	k_{25}	0.27
k_4	0.27	k_{15}	0.0067	k_{26}	0.073
k_5	0.0017	k_{16}	0.0068	k_{27}	0.05
k_6	0.013	k_{17}	0.0032	c_1	1.0
k_7	0.0025	k_{18}	0.0038	c_2	0.2
k_8	0.017	k_{19}	0.0077	cRAF_{tot}	0.013
k_9	0.0022	k_{20}	0.0058	MEK_{tot}	1.2
k_{10}	0.049	k_{21}	0.039	ERK_{tot}	0.74
k_{11}	0.0082	k_{22}	0.021		
k_{12}	0.0076	k_{23}	0.02		

Model Selection and Parameter Estimation. The model variety $\mathcal{V}_{\mathcal{M}d}$ of the distributive model is defined by (76)–(78) and the equations obtained by setting the “distributive” column

of table 5.6 equal to zero. Denote the system defining $\mathcal{V}_{\mathcal{M}_d}$ as \mathbf{F} . The model variety $\mathcal{V}_{\mathcal{M}_p}$ of the processive model is defined by (79)–(81) and the equations obtained by setting the “processive” column of table 5.6 equal to zero.

The ambient dimension of $\mathcal{V}_{\mathcal{M}_d}$ is 16 since the coordinates that define $\mathcal{V}_{\mathcal{M}_d}$ include x_1, \dots, x_{12} , np-ERK, pY-ERK, pYpT-ERK, and the model parameter k_1 . All other parameters and variables are known constants. The ambient dimension for the processive model is 18 since it includes the added variables x_{13} and x_{14} .

Given data $\mathbf{y} = (\text{np-ERK}', \text{pY-ERK}', \text{pYpT-ERK}')$ define the data variety for the distributive model as:

$$\mathcal{V}_{\mathcal{D}_d} = \{(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in \mathbb{C}^{16} : \mathbf{z} = \mathbf{y}\}.$$

The data variety $\mathcal{V}_{\mathcal{D}_d}$ has dimension 13. The data used takes the form of 36 concentration measurements of three aggregate phosphoforms over a range of 12 EGF stimulation levels and obtained directly by the authors of [28]. The EGF output data is summarized in table 5.8. The data variety $\mathcal{V}_{\mathcal{D}_p}$ for the processive model is defined similarly. Moving forward computations will be for the distributive model only. Computations for the processive model will be similar. Information for both models will be recorded.

First one computes a NID of $\mathcal{V}_{\mathcal{M}_d}$ using Bertini [8]. $\mathcal{V}_{\mathcal{M}_d}$ consists of a one-dimensional complex algebraic set of degree 8. Similarly for the processive model, the model variety $\mathcal{V}_{\mathcal{M}_p}$ is a one-dimensional complex algebraic set of degree 11. Several variables are then intrinsically defined to save computation. Variables x_1, x_2, x_7, x_{11} , and x_4 are written in terms of the other variables. In addition $\mathcal{V}_{\mathcal{M}_d} \cap \mathcal{V}_{\mathcal{D}_d} = \emptyset$ and $\mathcal{V}_{\mathcal{M}_p} \cap \mathcal{V}_{\mathcal{D}_p} = \emptyset$ using Bertini [8]. Variables np-ERK, pY-ERK, and pYpT-ERK are intrinsically defined to save computation. Since $\mathcal{V}_{\mathcal{M}_d} \cap \mathcal{V}_{\mathcal{D}_d} = \emptyset$ and $\mathcal{V}_{\mathcal{M}_p} \cap \mathcal{V}_{\mathcal{D}_p} = \emptyset$, algorithm 2 instructs us to minimize the distance

between $(\mathcal{V}_{\mathcal{M}_d})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}_d})_{\mathbb{R}}$ and similiarly between $(\mathcal{V}_{\mathcal{M}_p})_{\mathbb{R}}$ and $(\mathcal{V}_{\mathcal{D}_p})_{\mathbb{R}}$ for each data point in order to perform model selection using the distributive and processive models.

TABLE 5.8. Summary of EGF Level output data. np-ERK, pY-ERK, and pTpY-ERK are measured as percentage of total ERK. Each of the 12 levels of EGF loading consists of three aggregate phosphoforms.

EGF loading	np-ERK %	pY-ERK %	pTpY-ERK %
0.0244140625 ng/mL	0.968688845401175	0.0273972602739726	0.00391389432485323
	0.970703125	0.015625	0.013671875
	0.946135831381733	0.0351288056206089	0.0187353629976581
0.048828125 ng/mL	0.951219512195122	0.043360433604336	0.00542005420054201
	0.97423887587822	0.0210772833723653	0.00468384074941452
	0.937662337662338	0.0441558441558442	0.0181818181818182
0.09765625 ng/mL	0.937313432835821	0.0477611940298507	0.0149253731343284
	0.958937198067633	0.0265700483091787	0.0144927536231884
	0.885135135135135	0.0777027027027027	0.0371621621621622
0.1953125 ng/mL	0.893700787401575	0.0708661417322835	0.0354330708661417
	0.921182266009852	0.0394088669950739	0.0394088669950739
	0.853582554517134	0.0965732087227414	0.0498442367601246
0.390625 ng/mL	0.760180995475113	0.0950226244343891	0.144796380090498
	0.831288343558282	0.0552147239263804	0.113496932515337
	0.791411042944785	0.0828220858895705	0.125766871165644
0.78125 ng/mL	0.535211267605634	0.131455399061033	0.3333333333333333
	0.64453125	0.08203125	0.2734375
	0.584837545126354	0.169675090252708	0.245487364620939
1.5625 ng/mL	0.535211267605634	0.131455399061033	0.3333333333333333
	0.64453125	0.08203125	0.2734375
	0.584837545126354	0.169675090252708	0.245487364620939
3.125 ng/mL	0.0923076923076923	0.292307692307692	0.615384615384615
	0.223300970873786	0.140776699029126	0.635922330097087
	0.146718146718147	0.258687258687259	0.594594594594595
6.25 ng/mL	0.0276497695852535	0.271889400921659	0.700460829493088
	0.120218579234973	0.202185792349727	0.677595628415301
	0.0773809523809524	0.1875	0.735119047619048
12.5 ng/mL	0.0159362549800797	0.294820717131474	0.689243027888446
	0.107981220657277	0.178403755868545	0.713615023474179
	0.0753768844221105	0.278894472361809	0.645728643216081
25 ng/mL	0.0304182509505703	0.254752851711027	0.714828897338403
	0.0867924528301887	0.166037735849057	0.747169811320755
	0.050561797752809	0.264044943820225	0.685393258426966
50 ng/mL	0.00819672131147541	0.278688524590164	0.713114754098361
	0.0646551724137931	0.193965517241379	0.741379310344828
	0.0450450450450451	0.18018018018018	0.774774774774775

“Squaring up” the polynomial system defining $\mathcal{V}_{\mathcal{M}d}$ is a necessary step in order to accurately construct the polynomial system from proposition 5.3.1. Squaring polynomial systems via randomization is discussed in §2.1.7. The codimension of $\mathcal{V}_{\mathcal{M}d}$ is $c = 16 - 1 = 15$. There exists a nonempty Zariski open subset $\mathcal{A} \subset \mathbb{C}^{15 \times 17}$ so that $\mathcal{V}_{\mathcal{M}d} \subseteq \mathcal{V}(\mathbf{A}\mathbf{F})$ for every $\mathbf{A} \in \mathcal{A}$. Therefore let $\mathbf{A} \in \mathbb{C}^{15 \times 17}$, whose entries are taken randomly from the complex unit circle, and set $\mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z}) = \mathbf{A}\mathbf{F}(\mathbf{a}, \mathbf{x}, \mathbf{z})$. The aim is to solve the optimization problem:

$$(82) \quad \begin{aligned} & \text{minimize} && \|\mathbf{z} - \mathbf{y}\|^2 \\ & \text{subject to} && (\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}} \cap \mathbb{R}_{\geq 0}^{16}. \end{aligned}$$

Fritz John Conditions. The Fritz John conditions are:

$$(83) \quad \mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z}) = \mathbf{0},$$

$$(84) \quad \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial a_1} \lambda_j = 0,$$

$$(85) \quad \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial x_i} \lambda_j = 0, \text{ for } 1 \leq i \leq 12,$$

$$(86) \quad (z_i - y_i)\lambda_0 + \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial z_i} \lambda_j = 0, \text{ for } 1 \leq i \leq 3,$$

where $\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \dots, \lambda_{15}] \in \mathbb{P}^{15}$. Equations (83)–(86) consist of 31 variables and equations defined on $\mathbb{C}^{16} \times \mathbb{P}^{15}$. In addition there exists a nonempty Zariski open subset $\mathcal{B} \subset \mathbb{C}^{16}$ so that for each $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_{15}) \in \mathcal{B}$ equations (83)–(86) may be defined using affine coordinates using a patch equation:

$$\alpha_0 \lambda_0 + \alpha_1 \lambda_1 + \dots + \alpha_{15} \lambda_{15} - 1 = 0.$$

The Fritz John conditions for optimality state that $(\bar{\mathbf{a}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ is a critical point of $\|\mathbf{z} - \mathbf{y}\|^2$ for $(\mathbf{a}, \mathbf{x}, \mathbf{z}) \in (\mathcal{V}_{\mathcal{M}})_{\mathbb{R}}$ if there is a $\bar{\boldsymbol{\lambda}} \in \mathbb{P}^{15}$ so that $(\bar{\mathbf{a}}, \bar{\mathbf{x}}, \bar{\mathbf{z}}, \bar{\boldsymbol{\lambda}})$ is a solution to equations (83)–(86). Critical points are then obtained by projecting onto $(\mathbf{a}, \mathbf{x}, \mathbf{z})$ using the mapping

$\pi(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = (\mathbf{a}, \mathbf{x}, \mathbf{z})$. Our approach computes every critical point from equations (83)–(86) and evaluates $\|\mathbf{z} - \mathbf{y}\|^2$ at each critical point.

We need to ensure that $x_1, \dots, x_{12}, a_1, z_1, z_2, z_3$ are nonnegative. That is, the model variety is replaced by $\mathcal{S}_{\mathcal{M}_d} = \mathcal{V}_{\mathcal{M}_d} \cap \mathbb{R}_{\geq 0}^{16}$. To minimize the distance between $\mathcal{S}_{\mathcal{M}_d}$ and $\mathcal{V}_{\mathcal{D}_d}$ using a NAG approach first solve the system (83)–(86) and then solve related systems by setting combinations of $x_1, \dots, x_{12}, a_1, z_1, z_2, z_3$ to zero. This approach was outlined in §5.4.2. Since the complex dimension of $\mathcal{V}_{\mathcal{M}_d}$ is one and $\mathcal{V}_{\mathcal{M}_d}$ does not contain any coordinate hyperplanes as components (i.e. where any coordinates of $(\mathbf{a}, \mathbf{x}, \mathbf{z})$ are zero), one can verify that $\mathcal{V}_{\mathcal{M}_d}$ restricted to every coordinate hyperplane is zero-dimensional. In this case, checking the boundary conditions of $\mathcal{V}_{\mathcal{M}_d} \cap \mathbb{R}_{\geq 0}^{16}$ becomes trivial.

Finding Critical Points. Consider the 36 data points where each consists of a triple $\mathbf{y} = (\text{np-ERK}', \text{pY-ERK}', \text{pYpT-ERK}')$ define by table 5.8. Our aim is perform model selection on the data points to select either the processive or distributive model. Rather than solve equations (83)–(86) independently for each data point instead define a homotopy $\mathbf{H} : \mathbb{C}^{32} \times \mathbb{C}^3 \rightarrow \mathbb{C}^{32}$:

$$(87) \quad \mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{p}) = \begin{bmatrix} \mathbf{f}^*(\mathbf{a}, \mathbf{x}, \mathbf{z}) \\ \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial a_1} \lambda_j \\ \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial x_i} \lambda_j, \quad \text{for } 1 \leq i \leq 12 \\ (z_i - p_i) \lambda_0 + \sum_{j=1}^{15} \frac{\partial f_j^*(\mathbf{a}, \mathbf{x}, \mathbf{z})}{\partial z_i} \lambda_j \quad \text{for } 1 \leq i \leq 3 \\ \alpha_0 \lambda_0 + \alpha_1 \lambda_1 + \dots + \alpha_{15} \lambda_{15} - 1 \end{bmatrix}$$

using a general parameter $\mathbf{p} = (p_1, p_2, p_3) \in \mathbb{C}^3$. When the parameter is specialized to $\mathbf{p} = \mathbf{y}$, the Fritz John conditions for a given optimization problem using data \mathbf{y} is recovered. Using theory of parameter homotopies (see §2.4) there is a nonempty Zariski open subset $\mathcal{P} \subseteq \mathbb{C}^3$ such that the number of nonsingular isolated roots of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{p})$ is maximal for $\mathbf{p} \in \mathcal{P}$.

Furthermore for any $\mathbf{p}^* \in \mathcal{P}$ every isolated root of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{y})$ may be obtained by constructing the straight-line homotopy $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{p}^*t + (1 - t)\mathbf{y})$ and tracking the one real-dimensional solution paths starting at the nonsingular isolated roots of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{p}^*)$ and obtaining the isolated roots of $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{y})$ as $t \rightarrow 0$. The advantages of employing a parameter homotopy is explained in §2.4. Concretely, applying a parameter homotopy amounts to a 58 times speed up for the distributive model and approximately a $100\times$ speed up for the processive model when a multihomogeneous structure is used.

In addition to employing a parameter homotopy solving scheme equations (83)–(86) have a natural homogeneous product structure. That is, after equations (83)–(86) are multihomogenized with respect to the product of projective spaces $\mathbb{P}^{16} \times \mathbb{P}^{15}$, where the first space corresponds to the coordinates $(\mathbf{a}, \mathbf{x}, \mathbf{z})$ and the second space corresponds to the coordinates $\boldsymbol{\lambda}$, the number of tracked paths is significantly reduced when compared to the space \mathbb{P}^{31} . Multihomogeneous homotopies are discussed in §2.5. Together with parameter homotopies an efficient homotopy is used to solve $\mathbf{H}(\mathbf{a}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}; \mathbf{p}^*)$ for $\mathbf{p}^* \in \mathcal{P}$.

One way to reduce further computation is to define variables that occur in (83)–(86) intrinsically. This is easily applied when variables can be expressed as a linear combination of other variables. Specifically we know from table 5.6 that:

$$x_2 = \text{cRAF}_{tot}$$

where cRAF_{tot} is constant in table 5.7. x_2 is then removed from explicit computation. That is, partial derivatives are no longer necessary with respect to the variable x_2 and x_2 is no longer defined explicitly when tracking homotopy paths.

Timing and Paths. Table 5.9 summarizes the sequence of reductions made in the number of paths by imposing a $\{(\mathbf{a}, \mathbf{x}, \mathbf{z}), \boldsymbol{\lambda}\}$ -homogeneous structure followed by intrinsically defining

TABLE 5.9. Path counts of models. ‘ $\{(\mathbf{a}, \mathbf{x}, \mathbf{z}), \boldsymbol{\lambda}\}$ -hom’ corresponds to $\{(\mathbf{a}, \mathbf{x}, \mathbf{z}), \boldsymbol{\lambda}\}$ variable grouping and ‘intrinsic x_2 ’ corresponds to the intrinsically defined x_2 .

	Total Degree	$(\mathbf{a}, \mathbf{x}, \mathbf{z}), \boldsymbol{\lambda}$ -hom	$\{(\mathbf{a}, \mathbf{x}, \mathbf{z}), \boldsymbol{\lambda}\}$ -hom + intrinsic x_2
Distributive Model	124,416 paths	3,744 paths	1,152 paths
Processive Model	248,832 paths	7,488 paths	2,304 paths

the variable x_2 along with the number of paths required using the standard total degree homotopy. See §2.2.3 for information about total degree homotopies.

Timing summaries for both the processive and distributive model can be found in table 5.10 . These timings include the NID required to compute the dimension of each pure-

TABLE 5.10. Expected timings for the MAPK model collected over 20 ‘random’ runs.

	Compute Dimension	Initial Solve (parallel)	Data Solve (all 36)
Distributive Model	4.50 sec \pm 0.53 sec	44.80 sec \pm 4.85 sec	27.80 sec \pm 3.16 sec
Processive Model	6.64 sec \pm 0.48 sec	91.67 sec \pm 7.69 sec	32.77 sec \pm 5.06 sec

dimensional component of the model variety \mathcal{V}_{M_d} and \mathcal{V}_{M_p} , computing the nonsingular solutions of (87) for the distributive model at a generic parameter $\mathbf{p}^* \in \mathcal{P}$ required to employ a parameter homotopy scheme (and a similar solve for the processive model), and the parameter homotopy to solve equations (83)–(86) for each data point (and a similar parameter homotopy for the processive model). Timings to compute the dimension of the model variety and the data solve were done in serial using a Apple MacBook Pro with 2.4 GHz Intel core i5 processor. The initial solves for the parameter homotopies were done in parallel using 96 (2.67 GHz Xeon-5650) compute nodes on the CentOS 5.11 operating system. The data solves were then done in serial using the same MacBook Pro.

Analyzing the output. Tables 5.11–5.12 record the distances between the data and model varieties for all 36 data points. A missing “interior” distance in tables 5.11–5.12 indicate there were no positive real critical points found on the interior of $\mathcal{V}_{M_d} \cap \mathbb{R}_{\geq 0}^{16}$ for the given

EGF level and replicate, for example. However we may still compute a distance to the boundary of the semi-algebraic set corresponding to each model. Distances are measured

TABLE 5.11. Distance to (smaller) distributive model variety.

EGF level	Replicate	“Interior” distance	“Boundary” distance
1	1	0.0025	0.0309
	2	0.0118	0.0266
	3	0.0145	0.0496
2	1	0.0024	0.0485
	2	0.0036	0.0249
	3	0.0130	0.0581
4	1	0.0098	0.0594
	2	0.0117	0.0377
	3	0.0218	0.1062
8	1	0.0221	0.0981
	2	0.0312	0.0714
	3	0.0259	0.1349
16	1	0.0870	0.2189
	2	0.0838	0.1559
	3	0.0814	0.1904
32	1	0.1243	0.4343
	2	0.1505	0.3374
	3	0.0791	0.3784
64	1	0.0388	0.6990
	2	0.1312	0.4648
	3	0.0473	0.5889
128	1	0.0959	0.8398
	2	0.0725	0.7501
	3	0.0594	0.7931
256	1	—	0.9093
	2	0.0427	0.8353
	3	—	0.8839
512	1	0.1291	0.9154
	2	—	0.8556
	3	0.0947	0.8597
1024	1	—	0.9111
	2	—	0.8817
	3	0.0970	0.8883
2048	1	—	0.9272
	2	—	0.8948
	3	—	0.9197

as averages across the three replicates at various EGF levels are summarized graphically in figure 5.3.

TABLE 5.12. Distance to (larger) processive model variety

EGF level	Replicate	“Interior” distance	“Boundary” distance
1	1	0.0176	0.0309
	2	0.0066	0.0266
	3	0.0183	0.0496
2	1	0.0281	0.0485
	2	0.0130	0.0249
	3	0.0247	0.0581
4	1	0.0282	0.0594
	2	0.0137	0.0377
	3	0.0421	0.1062
8	1	0.0379	0.0981
	2	0.0154	0.0714
	3	0.0514	0.1349
16	1	0.0284	0.2189
	2	0.0156	0.1559
	3	0.0246	0.1904
32	1	0.0392	0.4343
	2	0.0424	0.3374
	3	0.0561	0.3784
64	1	0.0735	0.6990
	2	0.0444	0.4648
	3	0.0717	0.5889
128	1	0.1218	0.8398
	2	0.0550	0.7501
	3	0.0899	0.7931
256	1	—	0.9093
	2	0.0557	0.8353
	3	—	0.8839
512	1	—	0.9154
	2	—	0.8556
	3	0.1149	0.8597
1024	1	—	0.9111
	2	—	0.8817
	3	0.1105	0.8883
2048	1	—	0.9272
	2	—	0.8948
	3	—	0.9197

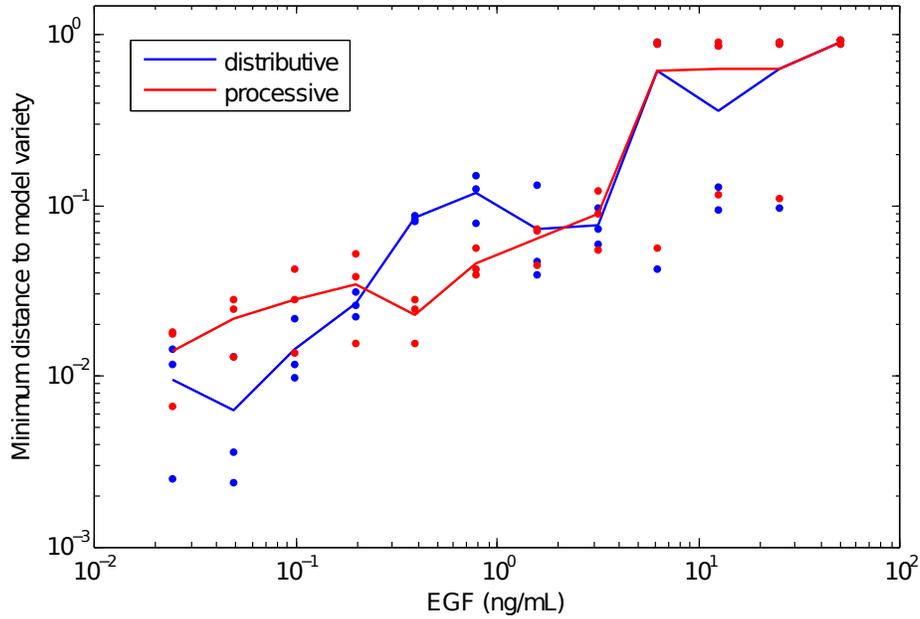


FIGURE 5.3. Distance plot for model selection between distributive and processive models.

One can interpret figure 5.3 as follows. Under low EGF stimulations the model selection estimates for d^2 are nearly identical with a slight preference for the distributive model. Under high EGF stimulation the models are nearly identical with no preference for one model over the other. The main difference between the distributive and processive models are the dynamics that model a nonlinear switching behavior that occur at intermediate EGF stimulation levels. At medium EGF stimulation, there is a slight preference to select the processive model. This supports the findings of [4].

CHAPTER 6

CONCLUSION

The aim of this thesis was to demonstrate how numerical algebraic geometry can be used to solve global optimization problems arising from applications in science and engineering. In chapter 2, I explained the fundamentals of numerical algebraic geometry that laid the foundation of the remaining chapters on perturbed regeneration, MLV line, and model selection. Development on the fundamentals of NAG is an activate area of research and there are still a wealth of challenging problems to solve.

Chapter 3 focused on perturbed regeneration. Perturbed regeneration is a technique to find all isolated solutions of a polynomial system including singular solutions. As we saw, this method performs well especially when singular solutions are desired, but there are also many other methods that may perform better. A future direction of study is to try and determine automatically what approach to take such as multihomogeneous or regeneration homotopies and their respective perturbed versions. In chapter 5, perturbed regeneration was then applied to solving the FJ conditions for optimality for model validation in the cluster model. Perturbed regeneration appears to work well in this context when it is unclear how the data and model varieties will intersect.

Chapter 4 focused on the max-length vector line of best fit to a collection of subspaces. Using numerical algebraic geometry we found all the critical points of an objective function involving principal angles between subspaces. Under a reformulation, we could define the critical points as solutions to the multivariate eigenvalue problem. By relaxing the condition of orthonormality, this method could be extended for finding the longest vector obtained by summing vectors from a collection of hyperellipsoids. The utility of the MLV line was

demonstrated on image data generated from the Pattern Analysis Lab and Colorado State University. The numerical algebraic geometry approach could be extended to other types of manifold means such as weighted max-length vector lines, k -dimensional subspaces of best fit, and flags of best fit, for example.

Finally in chapter 5 we focused on a new model selection paradigm. Here numerical algebraic geometry was applied to tackle three fundamental problems in science: model selection, model validation, and parameter estimation for polynomial dynamical systems. We highlighted models selection with experimental data in the MAPK/ERK pathway and showed for intermediate EGF levels were the most informative for model selection. It would be interesting to develop more computational efficient numerical algebraic geometry methods to handle inequality constraints as I believe this is the next step toward making NAG more applicable for global optimization.

BIBLIOGRAPHY

- [1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Appl. Math.*, 80(2):199–220, 2004.
- [2] Eugene L. Allgower, Daniel J. Bates, Andrew J. Sommese, and Charles W. Wampler. Solutions of polynomial systems derived from differential equations. *Computing*, 76(1–2):1–10, 2006.
- [3] James Anderson and Antonis Papachristodoulou. On validation and invalidation of biological models. *BMC Bioinformatics*, 10(1):132, 2009.
- [4] Kazuhiro Aoki, Masashi Yamada, Katsuyuki Kunida, Shuhei Yasuda, and Michiyuki Matsuda. Processive phosphorylation of ERK MAP kinase in mammalian cells. *Proc. Natl. Acad. Sci. U.S.A.*, 108(31):12675–12680, 2011.
- [5] Daniel J. Bates, Brent R. Davis, Elizabeth Gross, Heather Harrington, and Ken Ho. Numerical algebraic geometry for model selection and its application to the life sciences. *J. R. Soc. Interface*, 2016.
- [6] Daniel J. Bates, Brent R. Davis, Michael Kirby, Justin Marks, and Chris Peterson. The max-length-vector line of best fit to a set of vector subspaces and an optimization problem over a set of hyperellipsoids. *Numer. Linear Algebr.*, 22(3):453–464, 2015.
- [7] Daniel J. Bates, Jonathan D. Hauenstein, Chris Peterson, and Andrew J. Sommese. A numerical local dimension test for points on the solution set of a system of polynomial equations. *SIAM J. Numer. Anal.*, 47(5):3608–3623, 2009.
- [8] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for numerical algebraic geometry.(2015).

- [9] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25. SIAM, 2013.
- [10] Evgeni Begelfor and Michael Werman. Affine invariance revisited. *IEEE Comput. Soc. Conf. CVPR*, 2:2087–2094, 2006.
- [11] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions? *Int. J. Comput. Vision*, 28(3):245–260, 1998.
- [12] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions? *IEEE T. Pattern Anal.*, 23:643–660, 2001.
- [13] J. Ross Beveridge, Bruce A. Draper, Jen-Mai Chang, Michael Kirby, Holger Kley, and Chris Peterson. Principal angles separate subject illumination spaces in YDB and CMU-PIE. *IEEE T. Pattern Anal.*, 31(2):351–363, 2009.
- [14] Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comput.*, 27(123):579–594, 1973.
- [15] Grigoriy Blekherman, Pablo A. Parrilo, and Rekha R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.
- [16] Daniel A. Brake, Matthew Niemerg, and Daniel J. Bates. Paramotopy: parallel parameter homotopy through bertini. *Software available at www.paramotopy.com*, 2013.
- [17] Kenneth P. Burnham and David R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [18] John C. Butcher. An application of the runge-kutta space. *BIT*, 24(4):425–440, 1984.
- [19] Thomas C. Chamberlin. The method of multiple working hypotheses. *Science*, 148(3671):754–759, 1965.

- [20] Jen-Mei Chang, Michael Kirby, Holger Kley, Chris Peterson, Bruce Draper, and J. Ross Beveridge. Recognition of digital images of the human face at ultra low resolution via illumination spaces. In *Computer Vision-ACCV 2007*, pages 733–743. Springer, 2007.
- [21] Yasuko Chikuse. Procrustes analysis on some special manifolds. *Commun. Stat. Theory*, 28(3-4):885–903, 1999.
- [22] Moody T. Chu and J. Loren Watterson. On a multivariate eigenvalue problem, Part I: Algebraic theory and a power method. *SIAM J. Sci. Comput.*, 14(5):1089–1106, 1993.
- [23] Bruce Draper, Michael Kirby, Justin Marks, Tim Marrinan, and Chris Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra Appl.*, 451:15–32, 2014.
- [24] David Steven Dummit and Richard M. Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.
- [25] Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. A.*, 20(2):303–353, 1998.
- [26] P. Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on Riemannian manifolds with application to robust atlas estimation. *NeuroImage*, 45(1):S143–S152, 2009.
- [27] Simone Fulda and Klaus-Michael Debatin. Extrinsic versus intrinsic apoptosis pathways in anticancer chemotherapy. *Oncogene*, 25(34):4798–4811, 2006.
- [28] Alan S. Futran, A. James Link, Rony Seger, and Stanislav Y. Shvartsman. ERK as a model for systems biology of enzyme kinetics in cells. *Curr. Biol.*, 23(21):R972–R979, 2013.
- [29] Karin Gatermann. *Symbolic solution of polynomial equation systems with symmetry*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1990.

- [30] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry, 2002.
- [31] Andreas Griewank and Michael R. Osborne. Analysis of Newton’s method at irregular singularities. *SIAM J. Numer. Anal.*, 20:747–773, 1983.
- [32] Elizabeth Gross, Sonja Petrović, and Jan Verschelde. Interfacing with phcpack. *Journal of Software for Algebra and Geometry*, 5(1):20–25, 2013.
- [33] M. Hadjiandreou, Raul Conejeros, and Vassilis S. Vassiliadis. Towards a long-term model construction for the dynamic simulation of HIV infection. *Math. Biosci. Eng.*, 4(3):489–504, 2007.
- [34] Jihun Hamm. *Subspace-based learning with Grassmann kernels*. PhD thesis, University of Pennsylvania, 2008.
- [35] Wenrui Hao, Jonathan D. Hauenstein, Bei Hu, Yuan Liu, Andrew J. Sommese, and Yong-Tao Zhang. Continuation along bifurcation branches for a tumor model with a necrotic core. *J. Sci. Comput.*, 53(2):395–413, 2012.
- [36] Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 1977.
- [37] Jonathan D. Hauenstein. Numerically computing real points on algebraic sets. *Acta. Appl. Math.*, 125(1):105–119, 2013.
- [38] Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Regeneration homotopies for solving systems of polynomials. *Math. Comput.*, 80(273):345–377, 2011.
- [39] Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Regenerative cascade homotopies for solving polynomial systems. *Appl. Math. Comput.*, 218(4):1240–1246, 2011.

- [40] Jonathan D. Hauenstein and Charles W. Wampler. Isosingular sets and deflation. *Found. Comput. Math.*, 13(3):371–403, 2013.
- [41] Esteban A. Hernandez-Vargas, Dhagash Mehta, and Richard H. Middleton. Towards modeling HIV long term behavior. *IFAC Proceedings Volumes*, 44(1):581–586, 2011.
- [42] Kenneth L. Ho and Heather A. Harrington. Bistability in apoptosis by receptor clustering. *PLoS Comput. Biol.*, 6(10):e1000956, 2010.
- [43] Paul Horst. Relations among m sets of measures. *Psychometrika*, 26(2):129–149, 1961.
- [44] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Commun. Pur. Appl. Math.*, 30(5):509–541, 1977.
- [45] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.
- [46] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *Computer Vision–ECCV 2006*, pages 251–262. Springer, 2006.
- [47] Paul Kirk, Thomas Thorne, and Michael P. H. Stumpf. Model selection in systems and synthetic biology. *Curr. Opin. Biotech.*, 24(4):767–774, 2013.
- [48] Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi, et al. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [49] Anton Leykin, Jan Verschelde, and Ailing Zhao. Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in Algebraic Geometry*, pages 79–97. Springer, 2008.
- [50] Tien-Yien Li, Tim Sauer, and James A. Yorke. The cheater’s homotopy: an efficient procedure for solving systems of polynomial equations. *SIAM J. Numer. Anal.*, 26(5):1241–1251, 1989.

- [51] Juliane Liepe, Paul Kirk, Sarah Filippi, Tina Toni, Chris P. Barnes, and Michael P. H. Stumpf. A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. *Nat. Protoc.*, 9(2):439–456, 2014.
- [52] Gabriele Lillacci and Mustafa Khammash. Parameter estimation and model selection in computational biology. *PLoS Comput. Biol.*, 6(3):e1000696, 2010.
- [53] Alan E. Lindsay, Wenrui Hao, and Andrew J. Sommese. Vibrations of thin plates with small clamped patches. *P. R. Soc. A*, 471(2184), 2015.
- [54] Yui-Man Lui, J. Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *2010 IEEE Conference on CVPR*, pages 833–839. IEEE, 2010.
- [55] Tim Marrinan, J. Ross Beveridge, Bruce Draper, Michael Kirby, and Chris Peterson. Finding the subspace mean or median to fit your need. In *Proceedings of the IEEE Conference on CVPR*, pages 1082–1089, 2014.
- [56] Dhagash Mehta, Yang-Hui He, and Jonathan D. Hauenstein. Numerical algebraic geometry: a new perspective on gauge and string theories. *J. High Energy Phys.*, 7:1–32, 2012.
- [57] Pascal Meier, Andrew Finch, and Gerard Evan. Apoptosis in development. *Nature*, 407(6805):796–801, 2000.
- [58] Bishwarup Mondal, Satyaki Dutta, and Robert W. Heath. Quantization on the Grassmann manifold. *IEEE T. Signal Proces.*, 55(8):4208–4216, 2007.
- [59] Alexander P. Morgan and Andrew J. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Appl. Math. Comput.*, 24(2):115–138, 1987.

- [60] Alexander P. Morgan and Andrew J. Sommese. A homotopy for solving general polynomial systems that respect m -homogeneous structures. *Appl. Math. Comput.*, 24:101–113, 1987.
- [61] Alexander .P. Morgan and Andrew J. Sommese. Coefficient-parameter polynomial continuation. *Appl. Math. Comput.*, 29(2):123–160, 1989.
- [62] Ian Morrison and David Swinarski. Can you play a fair game of craps with a loaded pair of dice? *Am. Math. Monthly*, 123(2):136–148, 2016.
- [63] Takeo Ojika. Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations. *J. Math. Anal. Appl.*, 123(1):199–221, 1987.
- [64] Takeo Ojika, Satoshi Watanabe, and Taketomo Mitsui. Deflation algorithm for the multiple roots of a system of nonlinear equations. *J. Math. Anal. Appl.*, 96(2):463–479, 1983.
- [65] Inam U. Rahman, Iddo Drori, Victoria C. Stodden, David L. Donoho, and Peter Schröder. Multiscale representations for manifold-valued data. *Multiscale Model. Sim.*, 4(4):1201–1232, 2005.
- [66] Philipp Rostalski, Ioannis A. Fotiou, Daniel J. Bates, A. Giovanni Beccuti, and Manfred Morari. Numerical algebraic geometry for optimal control applications. *SIAM J. Optimiz.*, 21(2):417–437, 2011.
- [67] Hythem Sidky, Jonathan K. Whitmer, and Dhagash Mehta. Reliable mixture critical point computation using polynomial homotopy continuation. *AIChE J.*, 62:4497–4507, 2016.
- [68] Andrew J. Sommese, Jan Verschelde, and Charles W. Wampler. Advances in polynomial continuation for solving problems in kinematics. *J. Mech. Design*, 126(2):262–268, 2004.

- [69] Andrew J. Sommese and Charles W. Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005.
- [70] Andrew J. Sommese and Charles W. Wampler. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20:469–567, 2011.
- [71] Anuj Srivastava and Eric Klassen. Monte carlo extrinsic estimators of manifold-valued parameters. *IEEE T. Signal Proces.*, 50(2):299–308, 2002.
- [72] Bernd Sturmfels, Jonathan D. Hauenstein, and Jose Rodriguez. Maximum likelihood for matrices with rank constraints. In *ISSAC*, page 17, 2014.
- [73] Craig B. Thompson. Apoptosis in the pathogenesis and treatment of disease. *Science*, 267(5203):1456, 1995.
- [74] Pavan Turaga, Ashok Veeraraghavan, Anurag Srivastava, and Rama Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE T. Pattern Anal.*, 33(11):2273–2286, 2011.
- [75] Katharine Turner. Cone fields and topological sampling in manifolds with bounded curvature. *Found. Comput. Math.*, 13(6):913–933, 2013.
- [76] Kush R. Varshney and Alan S. Willsky. Linear dimensionality reduction for margin-based classification: High-dimensional data and sensor networks. *IEEE T. Signal Proces.*, 59(6):2496–2512, 2011.
- [77] Jan Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM T. Math. Software*, 25(2):251–276, 1999.
- [78] Charles W. Wampler, Alexander P. Morgan, and Andrew J. Sommese. Complete solutions to the nine-point path synthesis problem for four-bar linkages. *J. Mech. Design*, 114:153–159, 1992.

- [79] Tiesheng Wang and Pengfei Shi. Kernel Grassmannian distances and discriminant analysis for face recognition from image sets. *Pattern Recogn. Lett.*, 30(13):1161–1165, 2009.
- [80] Lior Wolf and Amnon Shashua. Learning over sets using kernel principal angles. *J. Mach. Learn. Res.*, 4:913–931, 2003.
- [81] Lei-Hong Zhang and Moody T. Chu. On a multivariate eigenvalue problem: II. Global solutions and the Gauss-Seidel method. http://www4.ncsu.edu/~mtchu/Research/Papers/On%20the%20MEP%20II_04.pdf, 2009.
- [82] Lei-Hong Zhang and Li-Zhi Liao. An alternating variable method for the maximal correlation problem. *J. Global Optim.*, 54(1):199–218, 2012.