KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF ENGINEERING SCIENCE

Back to the Roots

Polynomial System Solving Using Linear Algebra

Philippe DREESEN

Promotor:
Prof. dr. ir. B. De Moor

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

September 2013

Back to the Roots

Polynomial System Solving Using Linear Algebra

Philippe DREESEN

Supervisor:
Prof. dr. ir. Bart De Moor
(Department of Electrical Engineering)

Members of the Examination Committee:
Prof. dr. ir. Paula Moldenaers, chair
(Department of Chemical Engineering)
Prof. dr. ir. Karl Meerbergen, assessor
(Department of Computer Science)
Prof. dr. ir. Johan Suykens, assessor
(Department of Electrical Engineering)
Prof. dr. ir. Sabine Van Huffel, assessor
(Department of Electrical Engineering)
Prof. dr. ir. Joos Vandewalle, assessor
(Department of Electrical Engineering)
Prof. dr. ir. Johan Schoukens, assessor
(Vrije Universiteit Brussel, Belgium)
Prof. dr. Bernard Hanzon, assessor
(University College Cork, Ireland)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

September 2013

© KU Leuven – Faculty of Engineering Science
Department of Electrical Engineering
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
Kasteelpark Arenberg 10,
B-3001 Heverlee (Belgium)

D/2013/7515/88
ISBN 978-94-6018-702-5

Contents

Preface

The last years have been a challenging and life-changing trip. The support and company of supervisors, students and colleagues, friends, family and loved ones have made the journey unforgettable. This thesis would have never existed without the help and support of them.

First of all, I want to express my gratitude to my supervisor, prof. Bart De Moor, for giving me the opportunity to pursue a PhD in his group. I consider myself lucky having worked on a topic that is in the center of his attention. It wasn't always easy to tackle some hard algebraic questions with his street-fighting approach to mathematics, but the results in this text stem from the many inspiring and unforgettable meetings and brainstorming sessions.

Because my work ultimately didn't focus on the Structured Total Least Squares problem, prof. Sabine Van Huffel acts in the examination jury as assessor instead of co-supervisor. I am very grateful for her interest in my work and the many corrections and suggestions for the thesis.

Prof. Joos Vandewalle helped to organize and shape the education of mathematical knowledge of generations of KU Leuven engineers. I considered it an honor to be his teaching assistant for the course on System Identification. I will also never forget his close commitment to students and education as I often had the privilege to observe during WIT POC meetings. Finally I am very thankful for his interesting comments and remarks on the text.

I want to thank Prof. Johan Schoukens for letting me be the responsible teaching assistant for the exercises for his part of the System Identification course at KU Leuven. During our meetings on many events he was always very interested in my work and often brought me into contact with people at VUB working on related problems. I am looking forward to work on challenges involving system identification and polynomials in Brussels!

Prof. Johan Suykens deserves a special thank you. Not only for serving in the reading committee, but also for providing me with useful feedback on paper drafts, and involving me in the work of Kris De Brabanter and Tillmann Falck on the application of LS-SVM's in System Identification. Most of all, I will never forget the many lunches with Johan and the ALMA gang, where conversation topics were ranging from science, sport, travel, research, teaching, society and philosophy to quantum-mechanics. They were a pleasure every time.

I want to thank prof. Karl Meerbergen for the suggestions throughout my PhD and for taking part in the jury. I am very grateful to prof. Bernard Hanzon for the pleasant scientific conversation about polynomials and systems theory on the day before the preliminary defense. Finally, I want to thank prof. Paula Moldenaers for chairing the jury.

By working on the same topic with Kim Batselier, discovering new things about polynomials was always a shared joy. At the same time there was always someone to share the frustrations with when something didn't work. Along the way Kim became a great friend. Without his friendship and support this PhD would not have been here.

Academia turn out to be places where you meet a lot of interesting people. I would like to thank the many students, colleagues and ex-colleagues for the many interesting talks, discussions, coffee breaks, SISTA weekends, the drinks in Leuven, cinema nights, dinners, conferences and the many other occasions. Trying to name all is impossible, but I have always enjoyed the times spent with Tillmann, Marco, Dries, Kris, Pieter, Maarten, Toni, Raf, Niels, Tom, Fabian, Carlos, Marcelo, Mauricio, Siamak, Rocco, Vilen, Raghvendra, Marko, Xinhai, Nico, Mariya, Diana, Anca, Laurent, Mathias, Anna, Maarten, Laurent, Anne, and many more — this list can go on for a while. A big thank you goes out to Ilse, Ida, Lut, Anne, John, Elsy and Wim, as well as Maarten and Liesbeth for the excellent support with all administration and IT questions. Financially, part of this PhD research was supported by a PhD scholarship granted by IWT-Vlaanderen (1/2009-12/2012).

Be it by sharing houses, having ALMA lunches, going on Iceland trips or having poker nights, the following friends have made the last years an unforgettable time: Anca and Arthur, Anna and Maarten, David and Els, Jochen, Liesbeth and Thomas, Liesje and Jeroen, Mariya and Nikola, Pieter, Steven, Stijn and Griet, Tristan and Nathalie, Ward and Caroline, the '501' guys and girls, and many many more.

I want to thank my mother for all her support in everything. Thank you very much for being there for me. I also thank my father and step-mother for their continuous interest in my work. My sisters deserve a special thanks for reminding me from time to time how boring mathematics really is. I want to thank my grandmother and my other grandparents who are not here anymore, but have in profound ways contributed to this result. My interest in engineering was sparked by spending most of my childhood in the middle of the wires, cables, industrial engines and the tools of my grandfather's work shop. I want to thank the rest of my family and family-in-law for the support and interest in my work.

Finally, I want to thank my wife Gülin. We went through exciting but hectic times in the last months. But when I was writing this text, you always managed to give me time and space. I want to express my sincerest gratitude for your encouragement, patience, support and love. Without you this text would have never been written. My daughter Maren deserves a big thanks for giving me another reason for not having to sleep during the nights I was writing this text, but most of all for cheering me up with her sweet smiles. It is a miracle to see you discover the world. I am happy to be a part of it.

Leuven, August, 2013.

Philippe Dreesen

Abstract

Polynomial system solving is a classical mathematical problem occurring in science and engineering. We return to the original algebraic roots of the problem of finding the solutions of a set of polynomial equations. Rather than approaching the problem from symbolic algebra, we review this task from the linear algebra perspective and show that interesting links with systems theory and realization theory emerge.

The system of polynomial equations is represented by a structured Macaulay coefficient matrix multiplied by a vector containing monomials. Two properties are of key importance in the null spaces of Macaulay coefficient matrices, namely the correspondence between linear (in)dependent monomials in the polynomials and the linear (in)dependent rows in the null space, and secondly, the occurrence of a monomial multiplication shift structure. Both properties are invariant and hence occur regardless of the specific numerical basis of the null space of the Macaulay matrix.

Based on these insights, two algorithms for finding the solutions of a system of multivariate polynomials are developed. The first algorithm proceeds by computing a basis for the null space of the Macaulay matrix. By exploiting the multiplication structure in the monomials, a generalized eigenvalue problem is derived in terms of matrices built up from certain rows of a numerically computed basis for the null space of the Macaulay matrix. The second procedure does not require the computation of a basis for the null space of the Macaulay matrix. Rather, it operates on certain columns of the Macaulay matrix and again employs the property that a set of monomials in the problem are linearly dependent on another set of monomials. By using a proper partitioning of the columns according to this separation into linearly independent monomials and linearly dependent monomials, the problem of finding the solutions is again formulated as an eigenvalue problem, in this case phrased using a certain partitioning of the Macaulay matrix. It is shown that this can be implemented in a numerically reliable manner using a (Q-less) QR decomposition. Furthermore, the generalization of the null space-based root-finding algorithm to the case of overconstrained systems is discussed. Several applications in system identification and computer vision are highlighted.

The developed solution methods bear a resemblance to the application of realization theory as encountered in systems theory and identification. We show that the null space of the Macaulay matrix can be interpreted as a state sequence matrix of an nD system realization. It turns out that the notions of the regular and singular parts of an nD descriptor system naturally correspond to the affine solutions and the solutions at infinity.

Nederlandse Samenvatting

Het oplossen van stelsels multivariate veeltermvergelijkingen is een klassiek wiskundig probleem dat opduikt in een brede waaier van wetenschappelijke disciplines en ingenieurswetenschappen. In plaats van het probleem aan te pakken met symbolische algebra, bekijken we het probleem vanuit het perspectief van lineaire algebra en tonen aan dat er interessante verbanden met systeemtheorie en realizatietheorie opduiken.

Het stelsel van veeltermvergelijkingen wordt voorgesteld door een gestructureerde Macaulay coëfficiëntenmatrix die vermenigvuldigd wordt met een vector die de monomen bevat. Twee kenmerken zijn van essentieel belang in de nulruimte van de Macaulay-matrix, namelijk de overeenkomst tussen lineaire (on)afhankelijke monomen in de veeltermen en de lineaire (on)afhankelijke rijen van de nulruimte, en ten tweede, de aanwezigheid van een multiplicatieve schuifstructuur in de monomen. Beide kenmerken zijn invariant en treden dus op onafhankelijk van de specifieke numerieke basis van de nulruimte van de Macaulay-matrix.

Op basis van deze inzichten worden twee algoritmes ontwikkeld om stelsels veeltermvergelijkingen op te lossen. Het eerste algoritme start met het berekenen van een basis voor de nulruimte van de Macaulay-matrix. Door de multiplicatieve schuifstructuur in de monomen uit te buiten, wordt een veralgemeend eigenwaardenprobleem afgeleid dat geschreven is in termen van matrices die zijn samengesteld uit zekere rijen van de numerieke basis voor de nulruimte van de Macaulay-matrix. De tweede procedure vereist geen berekening van een numerieke basis voor de nulruimte. In plaats hiervan wordt er geopereerd op bepaalde kolommen van de Macaulay-matrix om opnieuw de eigenschap dat sommige monomen lineair afhankelijk zijn van andere monomen. Wanneer de juiste partitionering van de kolommen, gebaseerd op de indeling tussen lineair onafhankelijke en lineair afhankelijke monomen, plaatsvindt, kan de taak van het zoeken van de oplossingen opnieuw geformuleerd worden als een eigenwaardenprobleem. Vervolgens wordt aangetoond dat bepaalde operaties hierin efficiënt geïmplementeerd kunnen worden door middel van een QR ontbinding van de Macaulay-matrix. Voorts wordt het nulruimte-gebaseerde algoritme veralgemeend naar het zoeken van oplossingen van over-gedetermineerde (ruizige) stelsels veeltermvergelijkingen. Toepassingen in systeemidentificatie en beeldverwerking worden besproken.

De ontwikkelde oplossingsmethodes zijn verwant met het toepassen van realizatietheorie, een discipline in systeemidentificatie en systeemtheorie. We tonen dat de nulruimte van de Macaulay-matrix kan geïnterpreteerd worden als een toestandssequentie van een multidimensionale systeemrealizatie. Het blijkt tenslotte dat de noties van reguliere en singuliere delen in de multidimensionale toestandsbeschrijving overeenstemmen met de affiene nulpunten en de oplossingen op oneindig, respectievelijk.

Symbols and Notation

$:=$	'is defined as', *e.g.,* $x := y$ means 'x is defined as y'
$=:$	'is defined as', *e.g.,* $y =: x$ means 'x is defined as y'
$\mathbb{N} := \{0, 1, 2, \ldots\}$	set of natural numbers (including 0)
$\mathbb{C}$	set of complex numbers
n	number of unknowns x_i
$x_1, x_2, \ldots, x_n$	unknowns (or variables)
x_0	homogenization variable
$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$	exponent vector of the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \ldots x_n^{\alpha_n}$
$\lvert\boldsymbol{\alpha}\rvert$	degree of exponent vector $\boldsymbol{\alpha}$
s	number of polynomials in system
$f_1, f_2, \ldots, f_s$	polynomials (non-homogeneous)
$f_1^h, f_2^h, \ldots, f_s^h$	homogenized polynomials
$\rho_1 \approx 0, \ldots, \rho_s \approx 0$	over-constrained (noisy) polynomials
$d_i,$ for $i = 1, \ldots, s$	degrees of the polynomials f_i, *i.e.,* $d_i := \deg(d_i)$
$d_\circ = \max(d_1, \ldots, d_s)$	maximal degree occurring in the polynomials f_i
$\left(x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)}\right)$	j-th (affine) solution of a system

$\boldsymbol{M}(d)$	Macaulay matrix of polynomial system for degree d (Sylvester block structured)
$p(d)$	number of rows of $\boldsymbol{M}(d)$
$q(d)$	number of columns of matrix $\boldsymbol{M}(d)$
$\boldsymbol{N}(d)$	Macaulay matrix for degree d having nested quasi-Toeplitz structure
$\boldsymbol{k}(d)$	multivariate Vandermonde monomial vector containing monomials of degrees 0 up to d
$\boldsymbol{k}\vert_{\boldsymbol{x}^{(j)}}$	evaluation of j-th solution $\left(x_1^{(j)}, \ldots, x_n^{(j)}\right)$ in the monomial basis vector $\boldsymbol{k}$
$\boldsymbol{K}(d)$	multivariate Vandermonde structured basis for the null space of $\boldsymbol{M}(d)$
$\boldsymbol{H}(d)$	canonical basis for the null space of $\boldsymbol{M}(d)$
$\boldsymbol{Z}(d)$	numerical basis for the null space of $\boldsymbol{M}(d)$
$\boldsymbol{W}(d)$	column compressed null space
$\boldsymbol{S}_1$	row-selection matrix that selects low-degree blocks in $\boldsymbol{Z}$ (either standard monomials or entire degree-blocks)
$g(x_1, \ldots, x_n)$	polynomial shift function used in eigenvalue problem
$\boldsymbol{S}_g$	row-selection matrix that selects the rows of $g(x_1, \ldots, x_n)\boldsymbol{S}_1\boldsymbol{K}$
$\boldsymbol{D}_g$	diagonal matrix of eigenvalues which are the evaluation of the roots at $g(x_1, \ldots, x_n)$
$\boldsymbol{T}$	matrix of eigenvectors
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{A}^T$	transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of matrix $\boldsymbol{A}$
$\boldsymbol{A}^+$	Moore-Penrose pseudo-inverse of matrix $\boldsymbol{A}$
$\mathrm{diag}\,(a, b, c)$	diagonal matrix having the elements a, b and c on the diagonal (and size 3×3)
$\mathrm{rank}\,(\boldsymbol{M}(d))$	rank of matrix $\boldsymbol{M}(d)$
$\mathrm{nullity}\,(\boldsymbol{M}(d))$	nullity of matrix $\boldsymbol{M}(d)$
$\mathrm{size}(\boldsymbol{A})$	size of matrix $\boldsymbol{A}$
$\dim(\cdot)$	dimension (of a space or set)
$\deg(f)$	(total) degree of a polynomial f
$\\|\cdot\\|$	norm of vector or matrix (operator norm)

m	number of solutions of a polynomial system
m_B	Bézout number (number of roots of a generic polynomial system)
m_a	number of affine roots of a polynomial system
m_∞	number of roots at infinity
$\partial_j\vert_{x^\star}$	differential functional evaluated in $x^\star$
$\boldsymbol{K}_a$	matrix containing the columns of $\boldsymbol{K}$ corresponding to the affine roots
$\boldsymbol{K}_\infty$	matrix containing the columns of $\boldsymbol{K}$ corresponding to the roots at infinity
$\boldsymbol{K}_1$	matrix containing the rows of $\boldsymbol{K}$ that correspond to standard monomials
d	degree of the Macaulay matrix $\boldsymbol{M}(d)$
d_c	degree at which nullity of $\boldsymbol{M}(d)$ stabilizes
$d^\star$	Macaulay degree $d^\star := \sum d_i - n + 1$
d_G	degree at which gap between affine roots and roots at infinity can be detected in $\boldsymbol{M}(d)$
$B(d)$	standard monomials of Macaulay matrix of degree d
$B^\star(d_G)$	affine standard monomials
$\boldsymbol{v}(k), \boldsymbol{v}(k,l)$	regular part of state of (nD) Attasi state space model
$\boldsymbol{w}(k), \boldsymbol{w}(k,l)$	singular part of state of (nD) Attasi state space model
$\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n$	action matrices (regular) of nD Attasi state space model
$\boldsymbol{E}_0, \boldsymbol{E}_1, \ldots, \boldsymbol{E}_n$	action matrices (singular) of nD Attasi state space model
$\boldsymbol{\Gamma}$	annihilator of Sylvester matrix or Macaulay matrix
$\mathbb{C}[x_1, \ldots, x_n]$	ring of polynomials with coefficients in $\mathbb{C}$
$\mathbb{C}[x_1, \ldots, x_n]_{\leq d}$	ring of polynomials with coefficients in $\mathbb{C}$ and total degree $\leq d$
$\mathbb{C}[x_0, \ldots, x_n]$	ring of polynomials (projective space)
$\mathbb{C}[x_0, \ldots, x_n]_d$	ring of total degree d polynomials
$I := \langle f_1, \ldots, f_s \rangle$	ideal generated by the polynomials $f_1, \ldots, f_s$
$I_{\leq d}$	subset of I containing the elements of total degree $\leq d$
I^h	homogenization of ideal I, *i.e.*, $I^h := \langle f_1^h, \ldots, f_s^h \rangle$
I^h_d	subset of I^h containing the elements of total degree d

Part I

Introduction and Foundations

Introduction 1

1.1 Problem Statement

1.1.1 Solving Polynomial Equations

In this thesis, we develop (numerical) linear algebra algorithms for solving zero-dimensional systems polynomial equations, or, equivalently, 'finding the roots of systems of polynomial equations'. Polynomial system solving is an old and central problem in mathematics. It underlies many applications in applied mathematics, science and engineering (Buchberger, 2001; Cox et al., 2005, 2007; Dickenstein and Emiris, 2005; Mora, 2003, 2005; Sturmfels, 2002). The related question of solving a polynomial optimization problem arises in many engineering applications where one is often interested in finding the 'best' (optimal) solution for a certain problem.

The area of mathematics concerned with polynomial algebra is called 'algebraic geometry'. The body of literature in algebraic geometry is vast, and much of it is of a highly theoretical and abstract nature. The branch concerned with computational algorithms for solving questions in algebraic geometry is called 'computational algebraic geometry' or 'computer algebra', and has become an important research field for the last 50 years. This thesis develops methods for solving systems of polynomial equations that are inspired on linear algebra and realization theory concepts.

One of the obstacles in (computational) algebraic geometry is that most of the literature is only accessible after intensive study of the subject and, therefore, often beyond the grasp of applied mathematicians and engineers. We strongly believe that bridging the gaps between applied mathematics and algebraic geometry is of paramount importance. An important aim is therefore presenting the results in a didactical and accessible framework.

Although the topic is currently mainly studied from a very theoretical point of view, the range of applications of polynomial algebra is virtually endless, stretching from polynomial optimization (Bleylevens et al., 2007; Hanzon and Jibetean, 2003; Lasserre, 2001; Parrilo, 2000), over the analysis of statistical aspects (Pistone et al., 2001), the analysis of kinematic problems (Emiris, 1994), digital signal processing and systems theory (Buchberger, 2001) to bioinformatics (Emiris and Mourrain, 1999a; Emiris et al., 2006; Pachter and Sturmfels, 2005).

1.1.2 Approach Taken in Thesis

The problem of solving a system of multivariate polynomial equations is approached from the linear algebra point of view, with some inspiration from realization theory. We will develop two algorithms for finding the solutions of a given zero-dimensional system, solely by making use of straightforward numerical linear algebra techniques, such as eigenvalue computations, singular value decompositions and QR decompositions, and, importantly, without requiring the dominating notion of Gröbner bases (Becker and Weispfenning, 1993; Buchberger, 1965), see also Appendix B.6.

Apart from avoiding the formulation of a Gröbner basis, there are several advantages of phrasing the problem as a linear algebra question.

- First, in computers numbers can only be represented and manipulated in finite precision, which demands a careful consideration of the numerical aspects involved. Gröbner basis computations are based upon infinite precision arithmetic and therefore employ rational numbers, often resulting in outputs having huge coefficients (*i.e.,* hundreds of digits). The numerical linear algebra approach allows for a careful consideration of the numerical aspects while using finite-precision arithmetic. The numerical aspects are well-known and can be controlled.

- Furthermore, a system of polynomial equations may be obtained from a noisy experimental setting, which requires taking into account the limited accuracy of the experiment. The numerical linear algebra framework provides us with well-established numerical tools for handling with such issues. For instance, our framework will allow for certain related generalizations which would become rather cumbersome in the symbolic case. An instance is the case of (noisy) overdetermined systems of multivariate polynomials, which do likely not have any exact solutions, but which may have approximate solutions.

- Finally, we believe that the framework of linear algebra is very powerful from the didactical point of view. An engineer or applied mathematician with a working knowledge of linear algebra will easily understand and is able to implement our methods.

The methods presented here do not always (or necessarily) outperform existing methods in computational algebraic geometry. Similar approaches to some of our algorithms have been described earlier, see Section 1.2. However, it is to the authors' knowledge the first time that the presented results have been collected and written down in this simple form, with the intention of remaining as close as possible to the familiar language and (numerical) implementations of linear algebra.

The current manuscript is an ideal starting point to get introduced to more technical computational algebraic geometry literature. We also believe that our work will open a whole new avenue of research challenges that may be tackled by applied mathematicians and engineers. To give an example, one of the aspects that is rather lacking in the current computer algebra literature is the notion of numerical robustness and conditioning. The framework presented here allows for the use of the well-known and well-studied methods of numerical algebra to answer such questions.

1.2 State of the Art

The current section gives an overview of the currently existing methods for solving systems of polynomial equations. We aim to give a concise overview and will focus on the solution methods *rooted* in linear algebra.

1.2.1 Univariate Root-finding

In this manuscript we study the problem of finding the solution to a system of multivariate polynomial equations. The most simple instance of this problem is to find the roots of a single polynomial equation in a single variable. Although sounding harmless, the problem of solving a univariate polynomial equation is a research field of its own, still studied today. As we will learn in the forthcoming chapters, some of the results of univariate polynomial algebra can easily be generalized to the multivariate case, whereas it becomes a lot more involved for other aspects.

Numerical Univariate Root-finding Methods

Pan (1997) gives an excellent overview of methods for solving a univariate polynomial equation. Although a host of different univariate root-finding methods exists, most of them are not of direct relevance for the remainder of this manuscript. There is however an important method we will briefly highlight, translating the univariate root-finding problem into an eigenvalue problem.

It is well-known that the eigenvalues of a matrix $\boldsymbol{A}$ correspond to the roots of its characteristic polynomial $p(\lambda) := \det(\boldsymbol{A} - \lambda \boldsymbol{I})$. The converse holds as well: The roots of an univariate polynomial $f(x)$ can be computed as the eigenvalues of its Frobenius companion matrix.

Theorem 1.1 (Frobenius companion matrix (Pan, 1997)). Consider an univariate polynomial $f(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_0$. Consider the matrix equation

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & & & 1 & 0 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-2} & -a_{n-1} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{pmatrix} x, \tag{1.1}$$

in which the matrix occurring in the left-hand-side of the equation is called the Frobenius companion matrix. The roots of $f(x)$ correspond to the eigenvalues of the Frobenius companion matrix.

Interestingly, it turns out that many univariate root-finding methods can in some way be interpreted as variations of the power iterations method (Golub and Van Loan, 1996) operating on the Frobenius companion matrix (Pan, 1997).

1.2.2 Multivariate Root-finding: Numerical Methods

A system of multivariate equations may be solved using a variation of Newton's method. This method is a good way to find a solution in the vicinity of a given initial guess. It is however generally not easy to guarantee that *all* solutions are found by executing Newton's method repeatedly.

Let $\boldsymbol{f} := (f_1, \ldots, f_n)^T$ contain n polynomials in n variables, such that $\boldsymbol{f}$ maps from $\mathbb{R}^n$ to $\mathbb{R}^n$. In this case, the Newton iteration can be written as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{J}_f^{-1}\left(\boldsymbol{x}^{(k)}\right) \boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right),$$

where $\boldsymbol{x} := (x_1, x_2, \ldots, x_n)^T$ and $\boldsymbol{J}_f\left(\boldsymbol{x}^{(k)}\right)$ denotes the Jacobian matrix evaluated at $\boldsymbol{x}^{(k)}$, where

$$\boldsymbol{J}_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \tag{1.2}$$

1.2.3 Multivariate Root-finding: Algebraic Methods

Resultants: Sylvester and Macaulay

In this thesis, we will study techniques that can be seen as a mix of resultant theory (Gelfand et al., 1994) and numerical linear algebra (Golub and Van Loan, 1996) — with a dash of systems theory (Kailath, 1998).

The notion of resultants can be traced back to an important paper by Sylvester (1853), who showed that checking whether two univariate polynomials have common roots is equivalent to checking whether a certain matrix built from the coefficients is singular (see Chapter 4).

Macaulay (1902, 1916) generalized Sylvester's results to the case of multivariate homogeneous polynomials. Sylvester and Macaulay discovered compelling results in polynomial algebra that are still of interest today (and especially to us) because of their intimate link to linear algebra interpretations.

Due to their inherent limiting computational complexity, many of the insights of Sylvester and Macaulay have been neglected during most of the 20th century, when the focus in algebraic geometry shifted away from polynomial system solving to abstract algebra.

Buchberger's Gröbner Bases

Only in the 1960's the computational aspects of algebraic geometry entered the scene again with the development of Buchberger's algorithm. This procedure computes a so-called Gröbner basis of a system of polynomial equations (Becker and Weispfenning, 1993; Buchberger, 1965).[1] The Gröbner basis approach has dominated computer algebra for the coming decades after its conception.

Although Gröbner bases were among the first efficient and useful algorithmic tools in algebraic geometry, it also has its shortcomings. A major disadvantage of Buchberger's algorithm is that operates by performing symbolic manipulations on the coefficients of the input equations. Its extension to floating point arithmetic is known to be rather cumbersome (Sasaki and Kako, 2010; Shirayanagi, 1993) with limited alternatives available (Jónsson and Vavasis, 2004; Stetter, 1997, 2004).

It must be noted that Gröbner bases have more applications than solving polynomials alone. Nevertheless, Gröbner bases are one of the major tools in solving polynomial systems. To some extent, it is therefore surprising that

[1] Buchberger named the result of his algorithm in honor of his PhD thesis advisor, Wolfgang Gröbner.

in much of the algebraic geometry literature, computing a Gröbner basis is perceived as the *result*, whereas we consider it as a mere *tool* for solving a system of polynomial equations. An interesting question is therefore whether the tools employed by Faugère (1999, 2002) may be used in the question of solving a system of polynomial equations, rather than computing a Gröbner basis of the system. To the best of the author's knowledge, this question has never been adequately addressed.

The methods of Faugère (1999, 2002) are currently the most efficient for computing a Gröbner basis. They formulate the problem as a linear algebra problem by finding a reduction of a large coefficient matrix. Although this approach allows for a great improvement of computing times for computing a Gröbner basis, many challenges remain. Most notably, the reductions/eliminations performed may be dramatically ill-posed.

Back to Linear Algebra: Lazard and Stetter

By the 1980's, the relevance of the work of Sylvester and Macaulay was rediscovered by Lazard and Stetter (and coworkers) who utilize Macaulay-like matrices for solving polynomial systems. The revival of these results from mathematical antiquity resulted in the seminal papers Auzinger and Stetter (1988, 1989); Lazard (1981, 1983).

The earliest of these works seems to stem from Lazard (1981). The simple observation that Buchberger's algorithm bears a lot of resemblance to the Gaussian elimination led to the work of Lazard (1981, 1983) who described the computation of a Gröbner basis as triangularizing a large Macaulay-like matrix built from the coefficients of the system. The work of Lazard re-sparked the interest in matrix-based methods for solving polynomial algebra problems. Not only was the link to matrix algebra recapitulated, but also a (premature) link to eigendecompositions was established.

Only a few years later, another important milestone in this regard was reached. Supposedly independent of Lazard's work, the link between polynomial system solving and eigenvalue decompositions was thoroughly established in two papers by Auzinger and Stetter (1988, 1989). Although the work of Stetter in later years (Möller and Stetter, 1995; Stetter, 2004) would focus more on the numerical and algebraic aspects of the computations in the quotient space, rather than the construction of the eigenvalue problem, the early work had strong ties to Macaulay-like coefficient matrices.

In his book (Stetter, 2004), the emphasis is on numerical aspects, mainly of empirical polynomials (*i.e.,* having noisy coefficients) and the numerical repercussions, whereas the subproblem of finding a suitable monomial basis seems to be not of particular interest. Indeed, only in one of the very last chapters, Stetter (2004, Chapter 10) addresses the problem of finding a suitable monomial basis, reaching the conclusion that a lot of work is to be expected in this field, mainly centering on border bases and empirical Gröbner bases.

Furthermore, Stetter (2004, Chapter 10, p. 411) observes that currently, the only way to obtain the basis for the quotient space using commonly available software is via Gröbner basis methods. Approaches where the symbolic steps to find a basis for the quotient space are executed by means of (numerical) linear algebra seem to have been abandoned in much of the literature, as well as the ways to avoid the need for explicitly computing the Gröbner basis.

Stetter claims that the fact that presently only the Gröbner bases approach is the only available approach is the main reason why *polynomial algebra* has not been widely recognized in scientific computing so far. In Stetter (1996) he states that

> [. . .] matrix eigenproblems are not just *some* tool in the solution of polynomial systems of equations, but [. . .] they represent the *weakly nonlinear nucleus* to which the original, strongly nonlinear task may be reduced.

The 'Stetter approach' generally breaks down to two problems:

1. Finding a basis for the polynomial ring modulo the ideal generated by the input equations.

2. Expressing multiplication in the quotient space by means of multiplication matrices.

It can be shown that the eigenvalue decomposition of the multiplication matrix provides the (affine) solutions of the system. The solutions can then be obtained either from the eigenvalues, or from the eigenvectors. In Appendix B.6 we illustrate this procedure by making use of Gröbner basis computations.

The research efforts initiated by Lazard and Stetter were further explored by Corless et al. (1995); Emiris and Mourrain (1999b); Faugère (1999, 2002); Hanzon and Jibetean (2003); Jónsson and Vavasis (2004); Manocha (1994); Mourrain and Pan (2000), among others.

Most of these matrix computation variations for solving polynomial systems are stemming from resultant theory, in particular the u-resultant of van der Waerden (1931). The method by Jónsson and Vavasis (2004) uses the u-resultant and immediately phrases the root-finding problem as an eigenvalue problem from the Macaulay-like matrix, after dismissing certain rows in order to obtain a square eigenvalue problem. The method by Corless et al. (1995) employs the u-resultant and uses an SVD procedure to find the solutions.

1.2.4 Hybrid Approaches

Homotopy continuation methods (Li, 1997; Verschelde, 1996) employ a mixture of algebraic and numerical tools to solve a system of polynomial equations. By means of algebraic techniques, a root-count is obtained (*e.g.*, using the BKK or multi-homogeneous Bézout bound (Cox et al., 2005; Sturmfels, 2002)). Then, a so-called start system is constructed that has the same number of roots as the system that needs to be solved, but of which the solutions are known in advance.

The homotopy method proceeds by tracking the solutions of the start system while a continuous deformation transforms the start system to the original system. During the deformation of the coefficients, the trajectories of the solutions are tracked via Newton's method. A potential risk is that continuation methods run into problems if a solution trajectory passes through a region of ill-conditioning.

For an introduction to the subject of numerical homotopy continuation methods, see Li (1997); Verschelde (1996). The software implementation of Verschelde (1999) is currently among the most competitive methods for solving polynomial systems.

1.2.5 Polynomial Optimization

The recent years have witnessed an increased research interest in polynomial system solving and optimization (Dickenstein and Emiris, 2005; Sturmfels, 2002), with a myriad of applications in applied mathematics, science and engineering, such as systems and control (Buchberger, 2001), bioinformatics (Emiris and Mourrain, 1999a; Pachter and Sturmfels, 2005), robotics (Emiris, 1994), and many more.

This ongoing research interest has yielded interesting recent developments in real algebraic geometry and polynomial optimization (Hanzon and Jibetean, 2003; Lasserre, 2001; Lasserre et al., 2012; Laurent and Rostalski, 2012; Parrilo, 2000; Shor, 1987; Shor and Stetsyuk, 1997) that outperform many of the classical methods.

The so-called sums-of-squares polynomial optimization methods are based on convex relaxations: in the case that the method successfully finishes, a lower bound for the objective is found, which very often agrees with the optimum of the objective criterion.

In recent years the topic has received a lot of research attention, both for its theoretical beauty and its practical applications. Semidefinite programming problems can be solved in polynomial time, and the available implementations perform well in practice.

1.3 Research Objectives and Contributions

In the current section we give an overview of the research objectives and contributions of the thesis.

1.3.1 Linear Algebra and Realization Theory for Polynomial System Solving

Although from the state of the art overview it is clear that polynomial equations and linear algebra have common historical grounds, their intimate link has been neglected in most of the algebraic geometry literature since the end of the 19th century until well into the 20th century (also see Appendix C). The first objective of this thesis is therefore to phrase the problem of finding the solutions of a system of multivariate polynomial equations as a linear algebra problem.

The main contributions of this thesis are establishing conceptual links between polynomial system solving, linear algebra and realization theory. Chapter 3 contains some non-standard results from linear algebra and realization theory that will be used extensively throughout the text, such as solving homogeneous linear equations, computing a so-called canonical basis for the null space and the shift property that is prevalent in realization theory. These concepts will turn out to take up central roles in the interpretation of polynomial system solving as a linear algebra question. Appendix A provides the reader with the more basic results and more extensive background information about linear algebra and systems theory.

A link between univariate polynomials and linear dynamical systems arises naturally when considering difference equations and their characteristic polynomials. From linear system theory we know that the roots of the characteristic polynomial play a crucial role in describing and understanding the sequences that satisfy the corresponding difference equation. This is usually described by means of the Z-transform, see *e.g.,* Kailath (1998). Chapter 4 will work out these links for the univariate case with the main tool being the Sylvester matrix.

For multivariate polynomial systems this natural link arises as well, although only a limited amount of literature can be found on the subject. In the multivariate case the major tool in this interpretation is the Macaulay matrix, which is elaborately discussed in Chapter 5. The case of multivariate polynomials is described using multidimensional systems (so-called nD systems) as described by Attasi (1976); Bleylevens et al. (2007); Hanzon and Hazewinkel (2006).

Realization theory has been studied by Ho and Kalman (1966); Kung (1978); Willems (1986a,b, 1987) and has become a well-known tool in system identification, allowing to compute a state space model from a given set of input-output measurements of a system. In its simplest form the input-output measurements are stored in block Hankel matrices, from which a simple linear input-output relation can be written involving the observability matrix of the system and a state sequence matrix. A state space model description can then be retrieved using linear algebra techniques, *e.g.,* by SVD operations. For the case of multidimensional systems (so-called nD systems) similar techniques are available, see *e.g.,* Attasi (1976) and Gałkowski (2001).

Chapter 7 aims at exploring the links between multivariate polynomial system solving and nD realization theory. Using simplified nD models, we will show that polynomial system solving can be phrased as an nD realization problem involving descriptor systems, which can be decoupled into a regular and a singular part. In the context of polynomial equation solving the regular part can be associated to the affine solutions, whereas the singular part can be associated to the solutions at infinity.

The relevant publications for this part are Dreesen et al. (2012b, 2013b)

1.3.2 Two Algorithms for Solving Systems of Polynomial Equations

The observation that the task at hand has a strong link to linear algebra and dynamical systems theory naturally leads to two numerical linear algebra methods for solving systems of polynomial equations. Two algorithms for solving systems of polynomial equations are discussed in Chapter 6.

The first method starts with constructing a sufficiently large Macaulay coefficient matrix of which a numerical basis for the null space is computed. A shift-invariance property in the null space that is due to its interpretation as monomials and their inherent multiplication-invariance properties leads to the formulation of an eigenvalue problem from which all solutions of the system can be retrieved.

The second method does not require the computation of a numerical basis of the null space of the Macaulay matrix. Rather, it operates on certain columns of the Macaulay matrix and exploits the property that a set of monomials in the problem are linearly dependent on another set of monomials. By using a proper partitioning of the columns according to this separation into linearly independent monomials (*i.e.,* the standard monomials) and linearly dependent monomials, the problem of finding the solutions can again be formulated as an eigenvalue problem, in this case phrased using a certain partitioning of the Macaulay matrix. It will be shown that this can be implemented in a numerically reliable manner using a (Q-less) QR decomposition.

As an application of the methods developed in this chapter we will highlight a central problem occurring in system identification, namely the structured total least squares problem, which can be solved by finding the roots of a system of polynomial equations.

The relevant publications for this part are Dreesen and De Moor (2009); Dreesen et al. (2012a,b,c).

1.3.3 Solving Over-constrained Systems of Polynomial Equations

In many engineering and applied mathematics applications, the result of an experiment might be interpreted as a noisy realization of a set of coefficients of an underlying exact system of polynomial equations. Often it is possible to perform many of such experiments, which naturally leads to over-constrained systems of polynomial equations. Over-constrained systems of polynomial equations consist of more equations than unknowns, and have generically no solutions. However, finding the approximate solutions of such systems is often of great interest.

The null space based polynomial system solving method developed in this thesis provides a natural way to deal with such problems. We will confine our focus to a specific subset of over-constrained systems, namely the ones arising from noisy realizations of an underlying well-constrained system of equations. In this case, the existence of approximate solutions is ensured and also several other algebraic properties can be transferred from the well-constrained case. We will investigate how the proposed method deals with over-constrained problems, and point out how further research can tackle the broad case of over-constrained systems.

The relevant publication for this part is Dreesen et al. (2013a).

1.4 Thesis Overview

The text is organized as follows.

In **Chapter 2** we give a collection of examples showing the problems we will study in the thesis and the typical solution methods. They will serve as amuse-bouches for the reader and provide the general approach taken in this thesis.

Chapter 3 provides the reader with some background notions of linear algebra and realization theory. In particular, we will discuss the aspects involving the solution of systems of homogeneous linear equations and point out a duality property that will be important in the remainder of the thesis. Also a rudimentary algorithm for computing the canonical null space of a matrix will be presented. The canonical null space reveals the linear independence of the rows in the null space, and as such it gives important insight in some algebraic aspects that will be used thoroughly in the manuscript. Finally we will discuss an important property that is present in a basis of monomials, but has also close links to realization theory. This shift property will ultimately lead to the formulation of an eigenvalue problem from which the solutions of a system of polynomial equations can be obtained.

In **Chapter 4** we will discuss the case of solving a system of univariate polynomial equations. First of all the so-called Sylvester matrix is built, the null space of which will be used to find the common solutions of the system of equations. The univariate case is a trivial specialization of the methodology we will develop in the remainder of the thesis, but it is the ideal way to get acquainted with our approach and see the tools at work on some small and easy to grasp problems.

Chapter 5 discusses the Macaulay matrix, which is the multivariate generalization of the Sylvester matrix, and some of its relevant properties. Of particular interest to us is its null space, which we will describe using the observation that each of the common solutions of a system of equations describes a vector in the null space. The case of solutions with multiplicity and solutions at infinity will also be discussed, providing us with the main ingredients to develop root-finding methods.

In **Chapter 6** we will then formulate two root-finding algorithms and apply them to several examples and highlight a few applications. Both algorithms will result in eigendecompositions from which the solutions can be retrieved. The first algorithm will phrase an eigenvalue problem using a numerically computed basis for the null space of the Macaulay matrix. The second algorithm operates on a repartitioned Macaulay matrix of which the QR-decomposition is taken; the eigenvalue problem is then phrased in terms of a selection of the R-part.

The fact that polynomial system solving has close links with realization theory will be used in several chapters. In **Chapter 7** we will highlight this observation. In particular, we will show that the root-finding problem implicitly defines a multivariate state-space model of which the state sequence corresponds to the rows of the null space of the Macaulay matrix. As such, we will identify the Macaulay matrix as the interface between the system of polynomial equations and the interpretation of the solutions via eigenproblems.

Chapter 8 applies the methods developed in the thesis to the task of (approximately) solving a system of overdetermined polynomial equations.

Such problems arise often in practice, but are not straightforward to solve using the classical computational algebraic geometry methods. Our methods provide a natural framework for solving such problems. We will discuss our observations and describe an application in computer vision.

Finally, in **Chapter 9** we will summarize the thesis and point out several open problems and possibilities for future research.

Background and non-essential material has been collected in the appendices. **Appendix A** provides the reader with an in-depth introduction and overview of the methods of linear algebra and realization theory. **Appendix B** is an overview of algebraic geometry definitions and results that are relevant for the thesis. We have deliberately chosen to try to formulate our results without requiring most of the classical notions of algebraic geometry; an aim of the thesis is to develop a numerical algebra framework for tackling the task at hand. A collection of historical notes regarding the problem of solving polynomial equations is given in **Appendix C**.

In Figure 1.1 we give a schematic overview of the thesis. The text is organized in three parts: Foundations, Polynomial System Solving and Closing. Chapters 1 (Introduction), 2 (Motivational Examples), 6 (Macaulay Formulation), 7 (Two Algorithms for System Solving) and 10 (Conclusions and Outlook) constitute the essential work of the thesis. The remaining chapters provide secondary results. Background information is contained in the appendices.

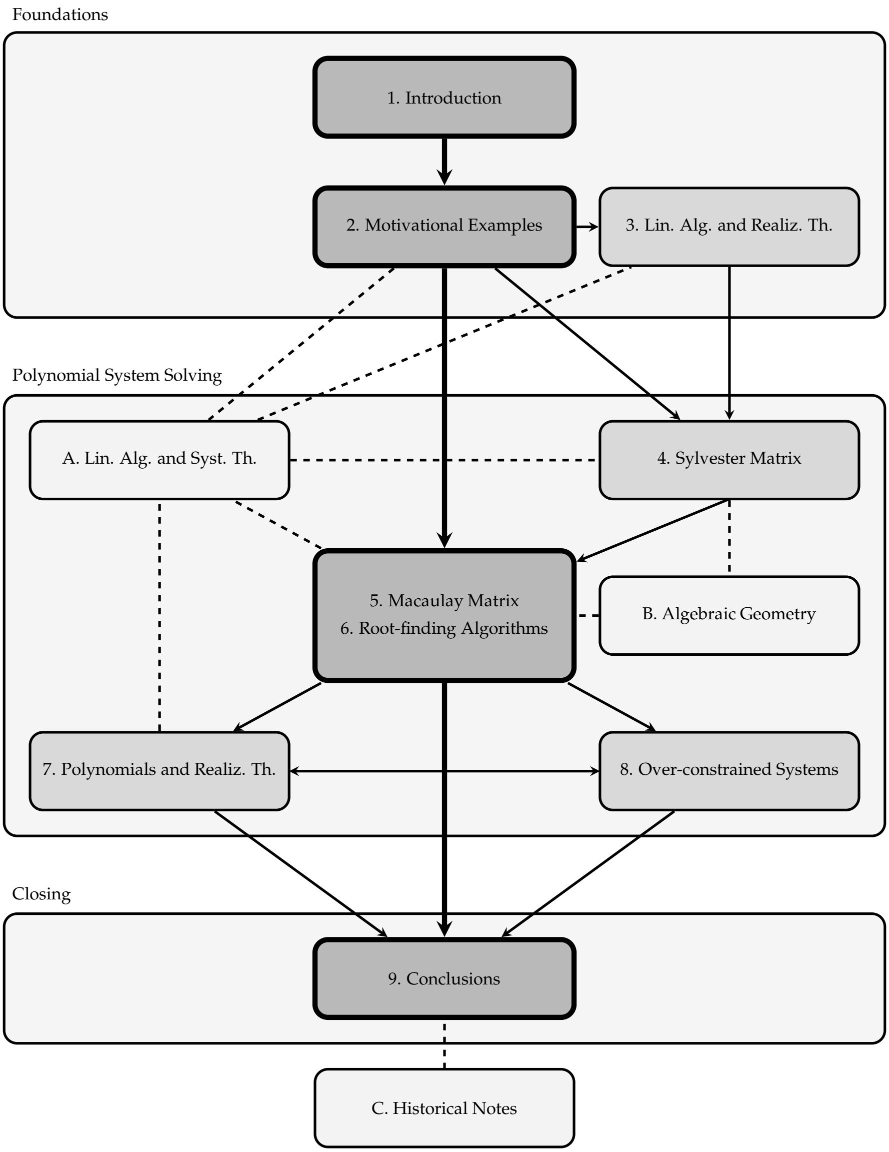
Foundations
1. Introduction
2. Motivational Examples
3. Lin. Alg. and Realiz. Th.
Polynomial System Solving
A. Lin. Alg. and Syst. Th.
4. Sylvester Matrix
5. Macaulay Matrix
6. Root-finding Algorithms
B. Algebraic Geometry
7. Polynomials and Realiz. Th.
8. Over-constrained Systems
Closing
9. Conclusions
C. Historical Notes

Figure 1.1: Flow chart representing the organization of the thesis chapters. The solid bold lines depict the connection between the essential chapters. The solid lines represent the links between remaining chapters. The connection between the appendices and the material in the chapters is visualized by the dotted line.

Motivational Examples 2

The current chapter is providing the reader with a bird's-eye view on the methodology that we will develop in the remainder of the thesis. By means of a small collection of didactical examples we will illustrate our procedures to solve systems of multivariate polynomial equations. We have aimed to make this chapter self-contained so as little as possible background knowledge is required to get an understanding of the methodology.

The underlying theory will be formalized in the forthcoming chapters, where emphasis will be placed on a more detailed study and general description as well as the treatment of numerical issues and implementation aspects.

2.1 Finding the Roots of Two Quadrics

2.1.1 Problem Formulation

Let us start with the following problem. We are given two polynomial equations in two variables x_1 and x_2 and we wish to compute the points of intersection, *i.e.,* the values of x_1 and x_2 that simultaneously fulfill the equations.

Consider the system

$$\begin{array}{rclcl} f_1(x_1,x_2) & = & -x_1^2+2x_1x_2+x_2^2+5x_1-3x_2-4 & = & 0, \\ f_2(x_1,x_2) & = & x_1^2+2x_1x_2+x_2^2-1 & = & 0, \end{array} \tag{2.1}$$

as visualized in Figure 2.1. The system has four real solutions (x_1,x_2): $(0,-1)$, $(1,0)$, $(3,-2)$ and $(4,-5)$, which we will try to determine using linear algebra tools.

In the next paragraphs we will develop a two-step approach to find the roots of the system (2.1). This approach will be the blueprint of the remainder of the procedures developed in manuscript.

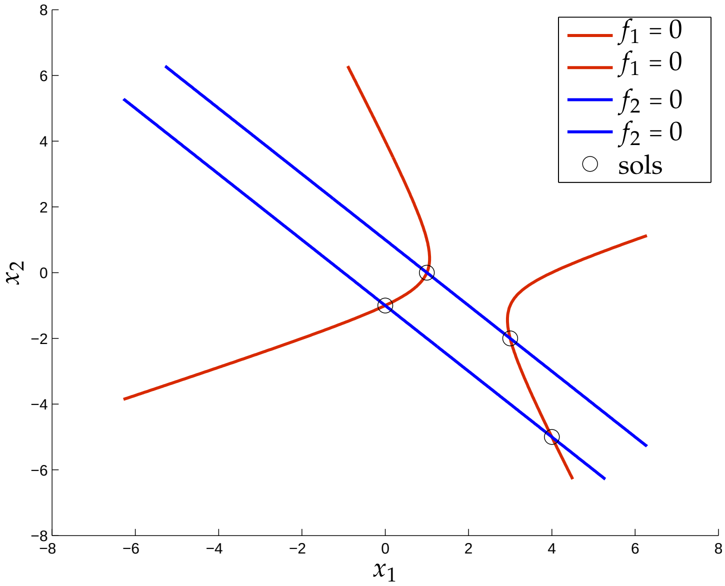

Figure 2.1: Graphical representation of the system given by the equations (2.1). There are four points for which both equations hold, namely the points $(0,-1)$, $(1,0)$, $(3,-2)$ and $(4,-5)$.

Step 1: The system of equations is considered as a set of linear homogeneous equations in the unknowns 1, x_1, x_2, x_1^2, x_1x_2, x_2^2. From this interpretation, we write the *dependent monomials* as a linear combination of the *independent monomials*. This step involves the construction of a coefficient matrix.

Step 2: By exploiting the multiplicative structure in the monomials we derive an eigenvalue problem from which the roots can be calculated. In this step we will make use of a basis for the null space of the coefficient matrix constructed in Step 1.

2.1.2 Two-Step Procedure for Finding the Roots

Step 1, iteration $d = 2$: Building the Macaulay Matrix

Since the input equations are of degree two, we let the iteration count start at $d = 2$. In iteration $d = 2$, we consider the two equations as a set of two homogeneous *linear* equations in the unknown monomials 1, x_1, x_2, x_1^2, x_1x_2,

x_2 as

$$\left(\begin{array}{c|cc|ccc} -4 & 5 & -3 & -1 & 2 & 1 \\ -1 & 0 & 0 & 1 & 2 & 1 \end{array}\right)\left(\begin{array}{c} 1 \\ \hline x_1 \\ x_2 \\ \hline x_1^2 \\ x_1x_2 \\ x_2^2 \end{array}\right) = \mathbf{0}.$$

or,

$$\boldsymbol{M}(2)\boldsymbol{k}(2) = \mathbf{0}.$$

In doing so, we have used the convention of ordering the monomials in the so-called degree negative lexicographic ordering (Definition 5.1), which for two variables is given as

$$1 < x_1 < x_2 < x_1^2 < x_1x_2 < x_2^2 < x_1^3 < x_1^2x_2 < x_1x_2^2 < x_2^3 < x_1^4 < \ldots$$

Using the coefficient matrix $\boldsymbol{M}(2)$, where 'M' stands for Macaulay and '2' represents the maximal degree of the monomials taken into account, the two equations are written in matrix-vector form. The rank of the coefficient matrix is two, hence its nullity (*i.e.,* the dimension of the null space: the number of columns of $\boldsymbol{M}$ minus the rank of $\boldsymbol{M}$) is four.

Since there are six unknowns, we can take four unknowns as independent variables, and two as dependent. The idea is that we try to take as dependent variables, the monomials that are as high in the ranking as possible.

Let us now inspect the columns of the coefficient matrix, starting from the right-most column. Clearly, the fifth column is linearly dependent on the sixth column. Column four is linearly independent of column six, so that the sub-matrix consisting of columns four and six is of rank two, hence allowing us to write the two dependent variables uniquely as a linear function of the remaining four variables.

The sub-matrix consisting of columns four and six of the coefficient matrix $\boldsymbol{M}(2)$ (*i.e.,* the columns corresponding to x_1^2 and x_2^2) is of rank two, we find x_1^2 and x_2^2 from the relation

$$\left(\begin{array}{cc} -1 & 1 \\ 1 & 1 \end{array}\right)\left(\begin{array}{c} x_1^2 \\ x_2^2 \end{array}\right) = -\left(\begin{array}{cccc} -4 & -3 & 5 & 2 \\ -1 & 0 & 0 & 2 \end{array}\right)\left(\begin{array}{c} 1 \\ x_1 \\ x_2 \\ x_1x_2 \end{array}\right).$$

We can now write the four solutions in the canonical matrix $\boldsymbol{H}(2)$, the columns of which form a basis for the null space of $\boldsymbol{M}(2)$. The rows of $\boldsymbol{H}(2)$ corresponding to $1, x_1, x_2$ and x_1x_2 form the identity matrix (bold-faced elements in the matrix) as

$$\boldsymbol{M}(2)\boldsymbol{H}(2) = \left(\begin{array}{c|cc|ccc} -4 & 5 & -3 & -1 & 2 & 1 \\ -1 & 0 & 0 & 1 & 2 & 1 \end{array}\right) \left(\begin{array}{cccc} \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ \hline -1.5 & 2.5 & -1.5 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 2.5 & -2.5 & 1.5 & -2 \end{array}\right) = \mathbf{0}$$

This terminates **iteration** $d = 2$ of **step 1**.

It can indeed be verified that $\boldsymbol{H}(2)$ is a basis for the null space of the Macaulay matrix $\boldsymbol{M}(2)$, however, it is not clear how it can be computed. Before we continue with the next steps of the root-finding procedure, we will show how to compute the *canonical null space* of the Macaulay matrix.

Intermezzo: Computing the Canonical Null Space

We will now briefly show how the canonical null space of the Macaulay matrix can be obtained. This method is inspired on Motzkin's double description method (Motzkin et al., 1953), seeking the non-negative solutions of linear systems. In Chapter 3 we will discuss the method elaborately.

The Motzkin procedure works on single rows as follows. The first row of the Macaulay matrix $\boldsymbol{M}(2)$ is

$$\boldsymbol{b}_1^T = \begin{pmatrix} -4 & 5 & -3 & -1 & 2 & \textcircled{1} \end{pmatrix},$$

of which a basis for the null space can easily be obtained by considering pair-wise combinations resulting in zero. We will call the right-most nonzero element (1) the pivot element. By forming all possible pair-wise eliminations between the pivot element and the remaining elements of $\boldsymbol{b}_1^T$, we obtain as a basis for the null space of $\boldsymbol{b}_1^T$ the matrix $\boldsymbol{H}_1$:

$$\boldsymbol{H}_1 = \left(\begin{array}{c|c|c|c|c} \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} \\ 4 & -5 & 3 & 1 & -2 \end{array}\right),$$

where every column prescribes one of the pair-wise eliminations. We have that $\text{rank}(\boldsymbol{H}) = 6 - 1$ and each column of $\boldsymbol{H}$ is orthogonal to $\boldsymbol{b}^T$, or $\boldsymbol{b}^T\boldsymbol{H} = \mathbf{0}$.[1]

[1] Special cases, such as the case that the right-most element is zero, or a whole row consists of zero elements are discussed in Chapter 3.

Due to the specific construction, the rows of the identity matrix sit at the top of $\boldsymbol{H}_1$.

By repeatedly applying this trick a complete basis for the null space of a given matrix is obtained. We proceed as follows. When the second row of $\boldsymbol{M}(2)$, denoted $\boldsymbol{b}_2^T$ is processed, it is first multiplied by $\boldsymbol{H}_1$ as to obtain $\boldsymbol{a}_2^T = \boldsymbol{b}_2^T \boldsymbol{H}_1$. We find

$$\begin{aligned} \boldsymbol{a}_2^T &= \boldsymbol{b}_2^T \boldsymbol{H}_1 \\ &= \begin{pmatrix} 3 & -5 & 3 & \textcircled{2} & 0 \end{pmatrix}. \end{aligned}$$

Now the above described procedure is performed on $\boldsymbol{a}_2^T$, giving us $\boldsymbol{H}_2$. We take as the pivot element the right-most nonzero element, *i.e.,* 2 and we find

$$\boldsymbol{H}_2 = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ -1.5 & 2.5 & -1.5 & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{pmatrix}.$$

Notice that a scaling is performed in order to ensure that the top part of $\boldsymbol{H}_2$ contains rows of the identity matrix in some of its rows.

Finally, the matrix $\boldsymbol{H} = \boldsymbol{H}_1 \boldsymbol{H}_2$ is a basis for the null space of $\boldsymbol{M}$ having the desired structure.[2] We have now

$$\boldsymbol{M}\boldsymbol{H} = \mathbf{0}$$

$$\left(\begin{array}{c|cc|ccc} -4 & 5 & -3 & -1 & 2 & 1 \\ -1 & 0 & 0 & 1 & 2 & 1 \end{array}\right) \left(\begin{array}{cccc} \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ \hline -1.5 & 2.5 & -1.5 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 2.5 & -2.5 & 1.5 & -2 \end{array}\right) = \mathbf{0}.$$

Step 1, iteration $d = 3$: Extending the Macaulay Matrix

We return to the solution of the system (2.1). The next iteration ($d = 3$) starts by multiplying each of the original two equations with the two first order monomials x_1 and x_2. This generates four more equations, each of degree three, reaching the additional monomials x_1^3, $x_1^2 x_2$, $x_1 x_2^2$ and x_2^3. Taking the two original equations together with these four *shifted* ones generates a set

[2] Indeed, by multiplying the consecutive rows with the previously obtained $\boldsymbol{H}_i$, it is guaranteed that $\prod \boldsymbol{H}_i$ is orthogonal to all previous rows of $\boldsymbol{M}$.

of six homogeneous linear equations in the ten unknown monomials up to degree $d = 3$, represented by

$$\left(\begin{array}{r|rr|rrr|rrrr} -4 & 5 & -3 & -1 & 2 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 & 0 \\ 0 & 0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{array}\right) \left(\begin{array}{c} 1 \\ \hline x_1 \\ x_2 \\ \hline x_1^2 \\ x_1x_2 \\ x_2^2 \\ \hline x_1^3 \\ x_1^2x_2 \\ x_1x_2^2 \\ x_2^3 \end{array}\right) = \mathbf{0}.$$

We denote the Macaulay coefficient matrix in this iteration as $\boldsymbol{M}(3)$. It is a 6×10 matrix, that contains as its rows the coefficients of the six equations $f_1 = 0, f_2 = 0, x_1f_1 = 0, x_2f_1 = 0, x_1f_2 = 0, x_2f_2 = 0$, and as its columns the coefficients of 1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, $x_1^2x_2$, $x_1x_2^2$ and x_2^3 in these equations.

It can be verified that $\text{rank}(\boldsymbol{M}(3)) = 6$, hence its nullity is $10 - 6 = 4$. Checking linear independence of the columns of $\boldsymbol{M}(3)$ starting from the right, we find that the columns x_1^2, x_2^2, x_1^3, $x_1^2x_2$, $x_1x_2^2$, x_2^3 are linear independent. The reason why we check the linear (in)dependency of the columns of $\boldsymbol{M}(3)$ from right to left is because we are using the complementarity property of Chapter 3: we wish to have in $\boldsymbol{H}(3)$ the top-most rows as the linearly independent rows.

The fact that the unknowns 1, x_1, x_2, x_1x_2 are the independent ones and the unknowns $x_1^2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3$ are dependent should come as no surprise: in iteration $d = 2$ we found that x_1^2 and x_2^2 are dependent variables, implying that also all monomials of higher degree that contain x_1^2 or x_2^2 as a factor, will be dependent.

The canonical basis for the null space of $\boldsymbol{M}(3)$, denoted by $\boldsymbol{H}(3)$, is given by

$$\boldsymbol{H}(3) = \left(\begin{array}{rrrr} \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ \hline -1.5 & 2.5 & -1.5 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 2.5 & -2.5 & 1.5 & -2 \\ \hline -3.75 & 4.75 & -3.75 & -1.5 \\ -3.75 & 3.75 & -3.75 & 5.5 \\ 11.25 & -11.25 & 11.25 & -9.5 \\ -18.75 & 18.75 & -17.75 & 13.5 \end{array}\right) \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right\} = \boldsymbol{H}(2) \\ \\ \\ \\ \\ \end{array}$$

in which the identity matrix sits at the position of the independent variables 1, x_1, x_2, x_1x_2, indicated by the bold-face numbers. Observe that the first 6 rows of $\boldsymbol{H}(3)$ are identical to those of $\boldsymbol{H}(2)$. Let us do one more iteration, which is **iteration** $d = 4$.

Step 1, iteration $d = 4$

We multiply the two original equations with the monomials x_1^2, x_1x_2, x_2^2, which generates another six shifted equations, this time fifteen monomials 1, x_1, x_2, …, x_1^4, …, x_2^4 are reached, resulting in the corresponding Macaulay matrix $\boldsymbol{M}(4)$:

$$\left(\begin{array}{c|cc|ccc|cccc|ccccc}
-4 & 5 & -3 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -4 & 0 & 5 & -3 & 0 & -1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -4 & 0 & 0 & 5 & -3 & 0 & 0 & -1 & 2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -4 & 0 & 0 & 5 & -3 & 0 & 0 & -1 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 5 & -3 & 0 & 0 & -1 & 2 & 1 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1
\end{array}\right).$$

Notice that $\boldsymbol{M}(2)$ and $\boldsymbol{M}(3)$ are 'nested' in the structure of $\boldsymbol{M}(4)$. Hence, one can obtain the subsequent matrices 'recursively' as the iteration number d increases.

The matrix $\boldsymbol{M}(4)$ is a 12×15 matrix, with $\operatorname{rank}(\boldsymbol{M}(4)) = 11$, so its nullity is four, as before. One can verify by monitoring linear independence of the columns starting from the right, that the columns corresponding to the monomials x_1^2, x_2^2, x_1^3, $x_1^2x_2$, $x_1x_2^2$, x_2^3, x_1^4, $x_1^3x_2$, $x_1^2x_2^2$, $x_1x_2^3$ and x_2^4 are linearly independent. This verifies that the variables 1, x_1, x_2 and x_1x_2 are still the independent variables, the other ones being dependent, which can also be seen from the canonical null space:

$$
H(4) = \left(\begin{array}{rrrr}
\mathbf{1} & 0 & 0 & 0 \\ \hline
0 & \mathbf{1} & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 \\ \hline
-1.5 & 2.5 & -1.5 & 0 \\
0 & 0 & 0 & \mathbf{1} \\
2.5 & -2.5 & 1.5 & -2 \\ \hline
-3.75 & 4.75 & -3.75 & -1.5 \\
-3.75 & 3.75 & -3.75 & 5.5 \\
11.25 & -11.25 & 11.25 & -9.5 \\
-18.75 & 18.75 & -17.75 & 13.5 \\ \hline
-1.5 & 2.5 & -1.5 & -12 \\
-26.25 & 26.25 & -26.25 & 26.5 \\
52.5 & -52.5 & 52.5 & -41 \\
-78.75 & 78.75 & -78.75 & 56.5 \\
107.5 & -107.5 & 106.5 & -74
\end{array}\right).
$$

For the subsequent iterations, where $d > 4$, the matrices $\boldsymbol{M}(d)$ and $\boldsymbol{H}(d)$ are too large to print, so we summarize the properties in the 'stabilization diagram' given in Table 2.1.

Table 2.1: Stabilization diagram for the system (2.1), showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d. The rank keeps increasing as d grows, however the nullity stabilizes at four. Also there are four linearly independent monomials that stabilize, in this example, right away from $d = 2$ onwards.

d	size $\boldsymbol{M}(d)$	rank $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	linearly independent monomials
2	2×6	2	4	$1, x_1, x_2, x_1x_2$
3	6×10	6	4	$1, x_1, x_2, x_1x_2$
4	12×15	11	4	$1, x_1, x_2, x_1x_2$
5	20×21	17	4	$1, x_1, x_2, x_1x_2$
6	30×28	24	4	$1, x_1, x_2, x_1x_2$

Notice that the number of rows of $\boldsymbol{M}(d)$ grows faster than the number of columns. Indeed, for degree d we have that the number of rows $p(d)$ and the number of columns $q(d)$ of $\boldsymbol{M}(d)$ is given by

$$p(d) = 2\binom{d}{d-2} = d^2 - d = \frac{d!}{(d-2)!},$$

and

$$q(d) = \binom{d+2}{d} = \tfrac{1}{2}d^2 + \tfrac{3}{2}d + 1 = \frac{(d+2)!}{2 \cdot d!}.$$

We also observe that the rank of $\boldsymbol{M}(d)$ keeps increasing as d grows, however, the nullity of $\boldsymbol{M}(d)$ stabilizes at the value four, which is the number of solutions. There are four linearly independent monomials that stabilize as well, being 1, x_1, x_2 and $x_1 x_2$. The general expression for the rank of $\boldsymbol{M}(d)$ in this example is given by $\mathrm{rank}(\boldsymbol{M}(d)) = \frac{1}{2}d^2 + \frac{3}{2}d - 3$.

Step 2: Finding the Roots

Let us now show how we can find the roots of the system of equations from the null space of the Macaulay matrix. In order to develop the root-finding technique, assume for the time being that we know the four true solutions $(0,-1)$, $(1,0)$, $(3,-2)$ and $(4,-5)$, which we denote as

$$\begin{array}{rcccl} \boldsymbol{x}^{(1)} & := & (x_1^{(1)}, x_2^{(1)}) & = & (0,-1), \\ \boldsymbol{x}^{(2)} & := & (x_1^{(2)}, x_2^{(2)}) & = & (1,0), \\ \boldsymbol{x}^{(3)} & := & (x_1^{(3)}, x_2^{(3)}) & = & (3,-2), \\ \boldsymbol{x}^{(4)} & := & (x_1^{(4)}, x_2^{(4)}) & = & (4,-5). \end{array}$$

Observe that each of the four solutions generates a vector in the basis of the null space of $\boldsymbol{M}(d)$. Indeed, evaluating the monomial basis vector

$$\boldsymbol{k}(d) = \left(\begin{array}{c|cc|ccc|c} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & \ldots \end{array} \right)^T$$

at each of the solutions essentially corresponds to evaluating the polynomials f_1 and f_2 at the solutions. By collecting these vectors in a matrix $\boldsymbol{K}(d)$ we find the multivariate Vandermonde basis of the null space of $\boldsymbol{M}(d)$ as (shown here for $d = 3$)

$$\begin{array}{rcl} \boldsymbol{K}(3) & = & \left(\begin{array}{cccc} | & | & | & | \\ \boldsymbol{k}(3)|_{\boldsymbol{x}^{(1)}} & \boldsymbol{k}(3)|_{\boldsymbol{x}^{(2)}} & \boldsymbol{k}(3)|_{\boldsymbol{x}^{(3)}} & \boldsymbol{k}(3)|_{\boldsymbol{x}^{(4)}} \\ | & | & | & | \end{array} \right) \\ & = & \left(\begin{array}{rrrr} 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 3 & 4 \\ -1 & 0 & -2 & -5 \\ \hline 0 & 1 & 9 & 16 \\ 0 & 0 & -6 & -20 \\ 1 & 0 & 4 & 25 \\ \hline 0 & 1 & 27 & 64 \\ 0 & 0 & -18 & -80 \\ 0 & 0 & 12 & 100 \\ -1 & 0 & -8 & -125 \end{array} \right). \end{array}$$

Let us now, starting from $\boldsymbol{H}(3)$ and the fact that 1, x_1, x_2 and x_1x_2 are the linear independent monomials of lowest degree, develop a method to find the roots $\boldsymbol{x}^{(i)}$, for $i = 1, \ldots, 4$. We will employ the multiplicative shift invariance property in the monomials of the multivariate Vandermonde vectors $\boldsymbol{k}(d)$. Let us consider a shift with the monomial x_1. We can write

$$\boldsymbol{S}_1 \boldsymbol{k} x_1 = \boldsymbol{S}_{x_1} \boldsymbol{k},$$

where $\boldsymbol{S}_1$ selects the rows 1, x_1, x_2, x_1x_2 from $\boldsymbol{k}$, and $\boldsymbol{S}_{x_1}$ selects the rows $1 \cdot x_1$, $x_1 \cdot x_1$, $x_2 \cdot x_1$, and $x_1x_2 \cdot x_1$ of $\boldsymbol{k}$. We have thus

$$\boldsymbol{S}_1 = \left(\begin{array}{c|cc|ccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

and

$$\boldsymbol{S}_{x_1} = \left(\begin{array}{c|cc|ccc|cccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

Applying this trick to the whole matrix $\boldsymbol{K}$ leads to the formulation of the generalized eigenvalue problem

$$\boldsymbol{S}_1 \boldsymbol{K} \boldsymbol{D}_{x_1} = \boldsymbol{S}_{x_1} \boldsymbol{K},$$

where $\boldsymbol{D}_{x_1} := \mathrm{diag}(x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_1^{(4)})$.

The multivariate Vandermonde matrix $\boldsymbol{K}$ reveals the four roots and their mutual matching, but it is not known on beforehand. Instead, we have computed the canonical basis as $\boldsymbol{H}(3)$. Let us now investigate how the two are related. We can easily verify that

$$\begin{array}{rcl} \boldsymbol{K} & = & \boldsymbol{H} \left(\begin{array}{cccc} 1 & 1 & 1 & 1 \\ x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & x_1^{(4)} \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & x_2^{(4)} \\ x_1^{(1)}x_2^{(1)} & x_1^{(2)}x_2^{(2)} & x_1^{(3)}x_2^{(3)} & x_1^{(4)}x_2^{(4)} \end{array}\right) \\ & = & \boldsymbol{H}\boldsymbol{T}, \end{array}$$

with $\boldsymbol{T}$ nonsingular.

Combining $\boldsymbol{K} = \boldsymbol{H}\boldsymbol{T}$ with $\boldsymbol{S}_1 \boldsymbol{K} \boldsymbol{D}_{x_1} = \boldsymbol{S}_{x_1} \boldsymbol{K}$ results in the generalized eigenvalue problem in the canonical null space

$$(\boldsymbol{S}_1 \boldsymbol{H}) \boldsymbol{T} \boldsymbol{D}_{x_1} = (\boldsymbol{S}_{x_1} \boldsymbol{H}) \boldsymbol{T},$$

where the matrix $\boldsymbol{D}_x$ contains the eigenvectors and the matrix $\boldsymbol{T}$ contains the eigenvectors.

Finally, by computing $\boldsymbol{HT}$ and scaling the result column-wise so that the first entries correspond to ones, we can reconstruct the multivariate Vandermonde structured basis. From this we can read off the x_1 and the corresponding x_2 components of the four roots.

2.1.3 Observations

This small example has taught us a lot about the interpretation of a polynomial system solving task as a question in linear algebra. Let us make the following important observations.

1. When using $\boldsymbol{H}$ as a basis for the null space of $\boldsymbol{M}$, the eigenvectors obey the multivariate Vandermonde structure. Moreover, $\boldsymbol{S}_1\boldsymbol{H}$ selects exactly the rows of $\boldsymbol{H}$ that contain the identity matrix rows: $\boldsymbol{S}_1\boldsymbol{H} = \boldsymbol{I}$.

2. When using the multivariate Vandermonde basis $\boldsymbol{K}$ to formulate the eigenvalue problem, the matrix $\boldsymbol{T}$ is the identity matrix: we have $\boldsymbol{S}_1\boldsymbol{KID}_{x_1} = \boldsymbol{S}_{x_1}\boldsymbol{KI}$.

3. We see that the choice of the basis only has an influence on the eigenvectors, but not on the eigenvalues: the matrix $\boldsymbol{D}_{x_1}$ is not changed by the choice of basis, only the eigenvectors change.

4. In principle, the matrix $\boldsymbol{S}_1$ may select more rows than only the linearly independent rows 1, x_1, x_2 and x_1x_2; in which case a *rectangular matrix pencil* will be obtained that is exactly solvable since the Vandermonde shift structure holds for all rows in $\boldsymbol{K}$. From the rectangular matrix pencil a square ordinary eigenvalue problem is found as

$$(\boldsymbol{S}_1\boldsymbol{H})^+\boldsymbol{S}_{x_1}\boldsymbol{H} = \boldsymbol{TD}_x\boldsymbol{T}^{-1},$$

   where $(\cdot)^+$ denotes the Moore-Penrose pseudo-inverse (see Chapter A). We will discuss the selection of more than only the linearly independent monomials rows in the forthcoming paragraphs.

5. We did not elaborate on the choice of d for which the eigenvalue problem was constructed. In this example, we have chosen $d = 3$ since the highest degree of the linearly independent monomials is two. Hence, shifting the linearly independent monomials with x_1 will require rows of degree at most three.

2.1.4 Using Other Shift Functions

Shift with x_2

The procedure can also be performed for a shift with the variable x_2. Again we let $\boldsymbol{S}_1$ select the rows of $\boldsymbol{K}$ corresponding to the rows 1, x_1, x_2, and x_1x_2. Now we define $\boldsymbol{D}_{x_2} := \mathrm{diag}(x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, x_2^{(4)})$ and $\boldsymbol{S}_{x_2}$ selects from $\boldsymbol{H}$ the rows corresponding to the multiplication of 1, x_1, x_2, x_1x_2 with the shift function x_2. We have

$$\boldsymbol{S}_1 = \left(\begin{array}{c|cc|ccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right),$$

and

$$\boldsymbol{S}_{x_2} = \left(\begin{array}{c|cc|ccc|cccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right).$$

The generalized eigenvalue problem revealing the roots x_2 is therefore

$$(\boldsymbol{S}_1\boldsymbol{H})\boldsymbol{T}\boldsymbol{D}_{x_2} = (\boldsymbol{S}_{x_2}\boldsymbol{H})\boldsymbol{T},$$

in which we observe that $\boldsymbol{S}_1\boldsymbol{K}$ is the same as for the shift with x_1. Moreover, the eigenvectors are also the same as in the case of using the shift x_1. An important consequence is that $(\boldsymbol{S}_1\boldsymbol{H})^{-1}(\boldsymbol{S}_{x_1}\boldsymbol{H})$ and $(\boldsymbol{S}_1\boldsymbol{H})^{-1}(\boldsymbol{S}_{x_2}\boldsymbol{H})$ commute. The commutation property should come as no surprise, as the multiplication of x_1 and x_2 is also commutative.

Shift with $g(x_1, x_2)$

The commutation property allows for using any polynomial function $g(x_1, x_2)$ as a shift function. Consider for example the polynomial

$$g(x_1, x_2) = 3x_1x_2 + 2x_2^2.$$

We now have

$$\begin{aligned} g((\boldsymbol{S}_1\boldsymbol{H})^{-1}(\boldsymbol{S}_{x_1}\boldsymbol{H}), (\boldsymbol{S}_1\boldsymbol{H})^{-1}(\boldsymbol{S}_{x_2}\boldsymbol{H})) &= 3\boldsymbol{T}\boldsymbol{D}_{x_1}\boldsymbol{T}^{-1}\boldsymbol{T}\boldsymbol{D}_{x_2}\boldsymbol{T}^{-1} \\ &\quad +2\boldsymbol{T}\boldsymbol{D}_{x_2}\boldsymbol{T}^{-1}\boldsymbol{T}\boldsymbol{D}_{x_2}\boldsymbol{T}^{-1}, \\ &= \boldsymbol{T}\left(3\boldsymbol{D}_{x_1}\boldsymbol{D}_{x_2} + 2\boldsymbol{D}_{x_2}\boldsymbol{D}_{x_2}\right)\boldsymbol{T}^{-1}, \\ &= \boldsymbol{T}\boldsymbol{D}_{g(x_1,x_2)}\boldsymbol{T}^{-1}. \end{aligned}$$

As a result, we can write the generalized eigenvalue problem as

$$\boldsymbol{S}_1 \boldsymbol{H} \boldsymbol{T} \boldsymbol{D}_g = \boldsymbol{S}_g \boldsymbol{H} \boldsymbol{T}, \tag{2.2}$$

where

$$\boldsymbol{D}_g := \operatorname{diag}\left(g\left(x_1^{(1)}, x_2^{(1)}\right), g\left(x_1^{(2)}, x_2^{(2)}\right), g\left(x_1^{(3)}, x_2^{(3)}\right), g\left(x_1^{(4)}, x_2^{(4)}\right)\right).$$

Naturally, when a shift polynomial $g(x_1, x_2)$ is considered, one needs to ensure that by shifting the linearly independent monomials with g, all monomials that are 'reached' are included in the (columns of) the Macaulay matrix. For instance, if the linearly independent monomials are 1, x_1, x_2, and $x_1 x_2$, and we wish to write the shift relation for $g(x_1, x_2) = 3x_1x_2 + 2x_2^2$, then the Macaulay matrix $\boldsymbol{M}(d)$ (and a basis for its null space) of degree $d \geq 2 + 2$ is required. On the other hand, the commutativity property prescribes that $\boldsymbol{D}_{g(x_1,\ldots,x_n)} = g(\boldsymbol{D}_{x_1}, \ldots, \boldsymbol{D}_{x_n})$, hence any shift $g(x_1, \ldots, x_n)$ can be composed by means of the monomial shifts.

2.1.5 About the Choice of Basis

It is important to realize that the 'multiplicative shift structure' is a property of the null space as a vector space, and not of the specific choice of basis. We will show that the derivation of the generalized eigenvalue problem holds for any arbitrary basis for the null space $\boldsymbol{Z}$, such as for instance a basis for the null space obtained using SVD. Let $\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{U}^{-1}$ with $\boldsymbol{U}$ a nonsingular matrix denote a basis for the null space of $\boldsymbol{M}$. We now have $\boldsymbol{H} = \boldsymbol{Z}\boldsymbol{U}$ and hence

$$\begin{aligned} \boldsymbol{S}_1 \boldsymbol{H}\boldsymbol{U}\boldsymbol{D}_x &= \boldsymbol{S}_x \boldsymbol{H}\boldsymbol{U}, \\ \boldsymbol{S}_1 \boldsymbol{H}\boldsymbol{U}\boldsymbol{D}_y &= \boldsymbol{S}_y \boldsymbol{H}\boldsymbol{U}, \end{aligned}$$

so we have

$$\begin{aligned} (\boldsymbol{S}_1 \boldsymbol{Z})(\boldsymbol{T}\boldsymbol{U})\boldsymbol{D}_x &= (\boldsymbol{S}_x \boldsymbol{Z})(\boldsymbol{T}\boldsymbol{U}), \\ (\boldsymbol{S}_1 \boldsymbol{Z})(\boldsymbol{T}\boldsymbol{U})\boldsymbol{D}_y &= (\boldsymbol{S}_y \boldsymbol{Z})(\boldsymbol{T}\boldsymbol{U}). \end{aligned}$$

Let us point out two important observations here.

1. The eigenvalues $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are not affected by the use of another basis for the null space.
2. The eigenvectors change, and they become $\boldsymbol{T}\boldsymbol{U}$.

2.1.6 Two Ways to Use the Shift Structure

Once we have computed a numerical basis for the null space of $\boldsymbol{M}(d)$, which we denote by $\boldsymbol{Z}$, the shift structure can be exploited in two ways, each of them giving rise to a generalized eigenvalue problem from which the solutions can be obtained.

1. The selection matrix $\boldsymbol{S}_1$ selects the linearly independent rows of $\boldsymbol{Z}$ only, leading to a square generalized eigenvalue problem in which it is ensured that the matrix $\boldsymbol{S}_1\boldsymbol{Z}$ is invertible. For our example we let $\boldsymbol{S}_1$ select the rows of $\boldsymbol{H}(3)$ corresponding to the monomials 1, x_1, x_2, x_1x_2. Let us now consider, for example, the shift function $g(x_1, x_2) = x_1 + 2x_2$ that maps these monomials to $x + 2y$, $x^2 + 2xy$, $xy + 2y^2$ and $x^2y + 2xy^2$, which is expressed by row selection and row combination matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_g$

$$\boldsymbol{S}_1 = \left(\begin{array}{c|cc|ccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right),$$

and

$$\boldsymbol{S}_g = \left(\begin{array}{c|cc|ccc|cccc} 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \end{array}\right).$$

We find the 4 × 4 matrices $\boldsymbol{S}_1\boldsymbol{H}$ and $\boldsymbol{S}_g\boldsymbol{H}$ from the eigenvalue problem as

$$\boldsymbol{S}_1\boldsymbol{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$\boldsymbol{S}_g\boldsymbol{H} = \begin{pmatrix} 0 & 2 & 3 & 0 \\ -3 & 5 & -3 & 3 \\ 7.5 & -7.5 & 4.5 & -4 \\ 26.25 & -26.25 & 26.25 & -17.5 \end{pmatrix},$$

and can solve the eigenvalue problem (2.2) from which we correctly retrieve the roots.

2. Alternatively, by letting $\boldsymbol{S}_1$ select possible all monomials of degree 2, such that the shifted monomials have the maximal degree occurring, *i.e.*, $d = 3$, we find the rectangular generalized eigenvalue problem

$$\boldsymbol{S}_1\boldsymbol{H}(3)\boldsymbol{T}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{H}(3)\boldsymbol{T}.$$

Since we know that $\boldsymbol{H}(3)$ (and hence $\boldsymbol{S}_1\boldsymbol{H}(3)$) has full column rank, we can rewrite the eigenvalue problem as the square ordinary eigenvalue problem

$$(\boldsymbol{S}_1\boldsymbol{H}(3))^+\boldsymbol{S}_g\boldsymbol{H}(3)\boldsymbol{T} = \boldsymbol{T}\boldsymbol{D}_g.$$

In our example we let $\boldsymbol{S}_1$ select all the rows of $\boldsymbol{H}$ that correspond to monomials of degrees zero up to three and then construct $\boldsymbol{S}_g$ in correspondence to S_1 and $g(x_1, x_2) = x_1 + 2x_2$. We have

$$S_1 = \left(\begin{array}{c|cc|ccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right),$$

and

$$S_g = \left(\begin{array}{c|cc|ccc|cccc} 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 \end{array}\right).$$

The matrices $S_g H(3)$ and $S_1 H(3)$ are in this case clearly of dimensions 10×4, and since $S_1 H(3)$ has full column rank, we can perform the eigenvalue decomposition on the square matrix

$$(S_1 H(3))^+ S_g H(3) = \begin{pmatrix} 0 & 2 & 3 & 0 \\ -3 & 5 & -3 & 3 \\ 7.5 & -7.5 & 4.5 & -4 \\ 26.25 & -26.25 & 26.25 & -17.5 \end{pmatrix}.$$

We see that we find exactly the same matrix as by selecting only the linearly independent rows.

As the eigenvalues we obtain the evaluation of the four roots in $g(x_1, x_2)$. The eigenvectors are used to recover the multivariate Vandermonde structure by computing $H(3)T$, where T contains the eigenvectors, and normalizing the columns such that the first row corresponds to $\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$. From the reconstructed Vandermonde structured basis we correctly find the four solutions

2.1.7 Conclusions

In this example we have developed the root-finding method based on eigendecompositions. Summarizing, to find the roots we first compute a basis for the null space of a Macaulay matrix M as Z. The selection of rows of Z corresponding to the linearly independent monomials results in $S_1 Z$. A shift function $g(x_1, x_2)$ is then chosen (*e.g.*, $g(x_1, x_2) = x_1$ or $g(x_1, x_2) = x_2$, or any polynomial function $g(x, y)$), which defines the selection matrix S_g and gives $S_g Z$. The generalized eigenvalue problem $S_1 Z T D_g = S_g Z T$ is solved and as the eigenvalues are returned the shift function $g(x_1, x_2)$ evaluated at the roots. We have observed the following properties:

- The row and column dimensions of the Macaulay matrix grow as a polynomial function as the iteration d for creating additional equations proceeds. Additional equations composing the Macaulay matrix are obtained by multiplying the original equations with all monomials of increasing degree.

- The rank of the Macaulay matrix increases, but the nullity stabilizes (in this example, the nullity was four right away, but typically, the nullity grows until it eventually stabilizes or keeps growing in a certain pattern). In this example, the set of linearly independent monomials stabilizes.

- The root-finding problem can be written as an eigenvalue problem. The most obvious way to do so is to consider the multivariate Vandermonde matrix, but it also holds for the canonical basis (the structure of which reveals the set of linearly independent monomials).

- The null space as a vector space exhibits three important invariants:

  1. the row-indices of the linear independent monomials do not change when another basis is considered,

  2. the multiplicative shift structure of the null space holds for all bases, and,

  3. the eigenvalues of the generalized eigenvalue problem do not depend on the specific choice of the basis for the null space.

- Any basis can be used to formulate the eigenvalue problem as the properties of the null space are universal. The specific choice of the basis for the null space of $\boldsymbol{M}(d)$ does not alter the eigenvalues, but only the eigenvectors.

- The matrices $(\boldsymbol{S}_1\boldsymbol{Z})^{-1}\boldsymbol{S}_{x_1}\boldsymbol{Z}$ and $(\boldsymbol{S}_1\boldsymbol{Z})^{-1}\boldsymbol{S}_{x_2}\boldsymbol{Z}$ commute and have common eigenspaces, which is the reason that any polynomial function $g(x_1, x_2)$ can be used as a shift function.

2.2 Three Equations in Three Unknowns

In the previous example, the equations were both of the same degree and the nullity of $\boldsymbol{M}(d)$ stabilized right away. The current example serves to illustrate two new points.

1. The first is that when not all equations are of the same degree, the initial Macaulay matrix should include also the so-called internal shifts of the equations of degrees lower than the maximal degree occurring in the system.

2. Secondly it is shown that that the nullity sometimes stabilizes only after a few degree-iterations.

2.2.1 Root-finding

Consider the equations

$$\begin{array}{lclcl} f_1(x_1,x_2,x_3) & = & x_1^2 - x_1x_2 + x_3 & = & 0, \\ f_2(x_1,x_2,x_3) & = & 2x_2^3 - 2x_1x_2^2 - 3x_1x_2 & = & 0, \\ f_3(x_1,x_2,x_3) & = & x_3^3 - x_1x_2x_3 - 2 & = & 0, \end{array} \tag{2.3}$$

where $d_1 = 2$ and $d_2 = d_3 = 3$.

We initiate the Macaulay matrix construction at degree $d = 3$. As in the previous section we consider the Macaulay matrix $\boldsymbol{M}(3)$ with columns indexed by all monomials up to degree three. Since equation f_1 is of degree two, we can also adjoin the shifted versions x_1f_1, x_2f_1 and x_3f_1 to the matrix $\boldsymbol{M}(3)$ so that we generate a maximum number of polynomials of degree three. This gives rise to the matrix $\boldsymbol{M}(3)$ as

$$\left(\begin{array}{r|rrr|rrrrrr|rrrrrrrrrr} 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 2 & 0 & 0 & 0 \\ \hline -6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 3 \end{array}\right),$$

of which the rows correspond to the polynomials (and their shifts) f_1, x_1f_1, x_2f_1, x_3f_1, f_2 and f_3 and the columns correspond to the monomials 1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, $x_1^2x_2$ $x_1^2x_3$, $x_1x_2^2$, $x_1x_2x_3$, $x_1x_3^2$, x_2^3, $x_2^2x_3$, $x_2x_3^2$ and x_3^3, which are again ordered by the degree negative lexicographic order (Definition 5.1).

The matrix size, nullity and the indices of the linearly independent monomials of $\boldsymbol{M}(d)$ for the consecutive degrees d are summarized in Table 2.2.

For degree d we have that the number of rows $p(d)$ and the number of columns $q(d)$ of $M(d)$ are given by the expressions

$$p(d) = \binom{d+1}{d-2} + 2\binom{d}{d-3} = \frac{(d+1)!}{2! \cdot (d-1)!} + 2\frac{d!}{3! \cdot (d-3)!} = \frac{1}{3}d^3 - d^2 + \frac{1}{2}d,$$

and

$$q(d) = \binom{d+3}{d} = \frac{(d+3)!}{3! \cdot d!} = \frac{1}{6}d^3 + d^2 + \frac{11}{6}d + 1.$$

In this example eighteen monomials 'stabilize', which is also the product of the degrees of the input equations. This is indeed no coincidence, and it will turn

Table 2.2: Stabilization diagram for the system (2.3) showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of degree d. The rank keeps increasing as d grows, however the nullity stabilizes at the value eighteen. Again, also the linear independent monomials stabilize.

d	size $\boldsymbol{M}(d)$	nullity $M(d)$	linearly independent monomials
3	6×20	14	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2$
4	18×35	17	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2, x_1^3x_3, x_1^2x_3^2, x_2^2x_3^2$
5	40×56	18	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2, x_1^3x_3, x_1^2x_3^2, x_2^2x_3^2, x_1^3x_3^2$
6	75×84	18	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2, x_1^3x_3, x_1^2x_3^2, x_2^2x_3^2, x_1^3x_3^2$
7	126×120	18	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2, x_1^3x_3, x_1^2x_3^2, x_2^2x_3^2, x_1^3x_3^2$
8	196×165	18	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, x_1^2x_3,$ $x_1x_3^2, x_2^2x_3, x_2x_3^2, x_1^3x_3, x_1^2x_3^2, x_2^2x_3^2, x_1^3x_3^2$

out (see Chapter 5) that the dimension of the null space of the Macaulay matrix corresponds to the so-called Bézout number $m_B = \prod_{i=1}^{n} d_i$ when the system has n equations in n unknowns and describes a zero-dimensional solution space (Cox et al., 2007). We will show in Chapter 5 that, for a sufficiently large degree d, we have

$$\text{nullity}(\boldsymbol{M}(d)) = \prod_{i=1}^{n} d_i,$$

where d_i denotes the degree of polynomial f_i.

The maximal degree occurring in the linearly independent monomials is five, so we set $d = 6$ and construct the Macaulay matrix — this will ensure that a linear shift will only require monomials that are included as columns of $\boldsymbol{M}(d)$. We compute a basis for the null space of $\boldsymbol{M}(6)$ as $\boldsymbol{H}(6)$ as in the previous example. We set up the generalized eigenvalue problem using a linear shift function $g(x_1, x_2, x_3)$ as in (2.2) and from the eigenvalues and eigenvectors we correctly retrieve the eighteen solutions (x_1, x_2, x_3) as

x_1	x_2	x_3
1.0000	3.0000	2.0000
-3.0019	-2.9256	-0.2291
-3.2091	-2.3900	-2.6284
-3.4075	-4.5860	4.0156
$-0.9721 \pm 0.5612i$	0.0000	$-0.6300 \pm 1.0911i$
$\pm 1.1225i$	0.0000	1.2599
$0.9721 \mp 0.5612i$	0.0000	$-0.6300 \pm 1.0911i$
$0.5407 \pm 0.4992i$	$-0.9250 \mp 0.1958i$	$-0.4456 \mp 1.1074i$
$-0.7163 \mp 0.2148i$	$-0.5872 \mp 1.5181i$	$-0.3725 \pm 0.9059i$
$-0.7163 \pm 0.2148i$	$-0.5872 \pm 1.5181i$	$-0.3725 \mp 0.9059i$
$-0.3041 \mp 0.8696i$	$-1.0230 \pm 0.5771i$	$1.4767 \pm 0.1852i$
$0.2891 \pm 0.6249i$	$1.4860 \pm 1.5589i$	$-0.2377 \pm 1.0179i$

2.2.2 Conclusions

The current example illustrates the following points.

1. When constructing the Macaulay matrix for the initial degree, it is sometimes necessary to bring the initial equations to the same degree as the maximal degree occurring in the original equations. This is done by multiplying the equations of lower degree with monomials up to $\max(d_i)$.

2. The nullity stabilizes only after a few iterations, together with all independent variables. The value corresponds to the Bézout number, which is defined as the product of the degrees of the equations.

3. The solutions of polynomials with real coefficients can be complex numbers occurring in complex conjugated pairs.

2.3 Roots at Infinity: *Mind the Gap!*

Let us now look at an example where the nullity stabilizes, but only some of the indices of the independent variables stabilize, and others do not. It turns out that the indices that do not stabilize can be explained by the so-called roots at infinity.

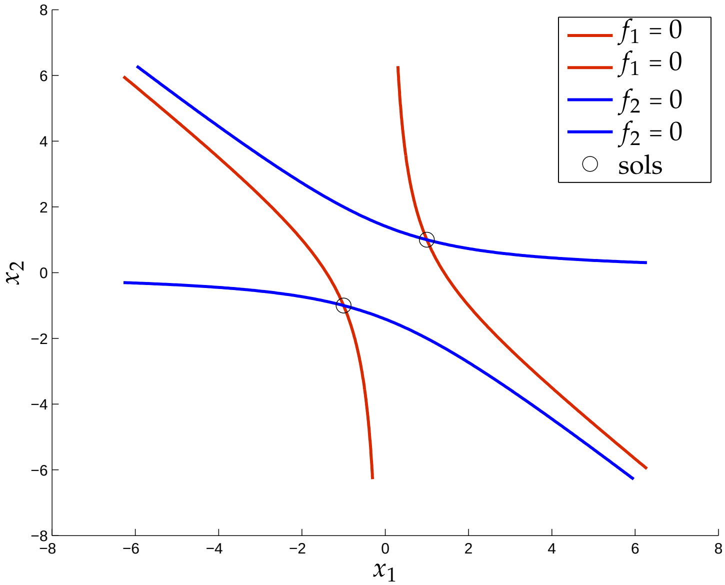

Figure 2.2: Graphical representation of (2.4). There are two solutions $(-1,-1)$ and $(1,1)$.

2.3.1 Root-finding

Consider the system of two equations

$$\begin{array}{rcccl} f_1(x_1,x_2) & = & x_1^2 + x_1x_2 - 2 & = & 0, \\ f_2(x_1,x_2) & = & x_2^2 + x_1x_2 - 2 & = & 0, \end{array} \tag{2.4}$$

which is shown in Figure 2.2. There are two solutions $(-1,-1)$ and $(1,1)$.

We construct for several iterations the Macaulay matrix and monitor its rank, nullity and the indices of the linearly independent monomials. The results are summarized in Table 2.3.

We observe that there are four linear independent monomials in all iterations, but only 1 and x_1 stabilize, while the other two monomials are replaced by higher degree monomials as d increases. There is a pattern in the two remaining monomials: they are always x_1^d and x_1^{d-1}.

The strange behavior of the linearly independent monomials can be explained by the fact that there are two roots at infinity. The two affine roots correspond to the monomials 1 and x_1, and two roots at infinity correspond to the monomials x_1^d and x_1^{d-1}. This can be understood from homogenizing the two

Table 2.3: Stabilization diagram for the system (2.4), showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d. The rank keeps increasing as d grows, however the nullity stabilizes at the value four. Observe that only two of the linear independent monomials stabilize, namely 1 and x_1 (indicated in bold-face), whereas the remaining two shift towards higher degrees as the overall degree of the Macaulay matrix increases.

d	size $\boldsymbol{M}(d)$	rank $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	linearly independent monomials
2	2×6	2	4	$1, x_1, x_2, x_1^2$
3	6×10	6	4	$\mathbf{1}, \mathbf{x_1}, x_1^2, x_1^3$
4	12×15	11	4	$\mathbf{1}, \mathbf{x_1}, x_1^3, x_1^4$
5	20×21	17	4	$\mathbf{1}, \mathbf{x_1}, x_1^4, x_1^5$
6	30×28	24	4	$\mathbf{1}, \mathbf{x_1}, x_1^5, x_1^6$

equations as

$$\begin{array}{rcccl} f_1^h(x_0, x_1, x_2) & = & x_1^2 + x_1 x_2 - 2x_0^2 & = & 0, \\ f_2^h(x_0, x_1, x_2) & = & x_2^2 + x_1 x_2 - 2x_0^2 & = & 0. \end{array}$$

By setting $x_0 = 0$, we can analyze the roots at infinity. We identify $x_1 + x_2$ as a common factor in both equations, which confirms that there exists a root at infinity $(x_0, x_1, x_2) = (0, 1, -1)$. As it turns out, this root has a double multiplicity, which explains the fact that we find two linearly independent monomials corresponding to it.

The existence of roots at infinity is also expressed in the Macaulay matrix. Indeed, if there can be found linear independent monomials of degree d in $\boldsymbol{M}(d)$, for any sufficiently large degree d, there are roots at infinity. It can be verified that setting the homogenization variable x_0 to zero in the homogenized system is equivalent to retaining only the highest degree columns of the Macaulay matrix. If there is linear dependence among these columns, there are roots at infinity.

The dynamical behavior of the structure of the null space when there are roots at infinity is also expressed in the canonical basis $\boldsymbol{H}(d)$, as d increases. At degree $d = 4$ we clearly see the separation emerging between the affine roots and the roots at infinity. At degree $d = 5$ the separation between the affine roots and the roots at infinity is increased by one degree block, as shown in

$$
\boldsymbol{H}(4) = \begin{array}{c}
\begin{array}{cccc} \multicolumn{2}{c}{\overbrace{\hphantom{0\;0}}^{\text{affine}}} & \multicolumn{2}{c}{\overbrace{\hphantom{0\;\;0}}^{\text{infinity}}} \end{array} \\
\left(\begin{array}{cc|cc}
\mathbf{1} & 0 & 0 & 0 \\ \hline
0 & \mathbf{1} & 0 & 0 \\
0 & 1 & 0 & 0 \\ \hline
1 & 0 & \mathbf{0} & \mathbf{0} \\
1 & 0 & \mathbf{0} & \mathbf{0} \\
1 & 0 & \mathbf{0} & \mathbf{0} \\ \hline
0 & 0 & \mathbf{1} & 0 \\
0 & 2 & -1 & 0 \\
0 & 0 & 1 & 0 \\
0 & 2 & -1 & 0 \\ \hline
0 & 0 & 0 & \mathbf{1} \\
2 & 0 & 0 & -1 \\
0 & 0 & 0 & 1 \\
2 & 0 & 0 & -1 \\
0 & 0 & 0 & 1
\end{array}\right)
\end{array}
\begin{array}{c} \\ \\ \\ \\ \uparrow \\ \text{gap} \\ \downarrow \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}
\quad \text{and} \quad
\begin{array}{c}
\begin{array}{cccc} \multicolumn{2}{c}{\overbrace{\hphantom{0\;0}}^{\text{affine}}} & \multicolumn{2}{c}{\overbrace{\hphantom{0\;\;0}}^{\text{infinity}}} \end{array} \\
\left(\begin{array}{cc|cc}
\mathbf{1} & 0 & 0 & 0 \\ \hline
0 & \mathbf{1} & 0 & 0 \\
0 & 1 & 0 & 0 \\ \hline
1 & 0 & \mathbf{0} & \mathbf{0} \\
1 & 0 & \mathbf{0} & \mathbf{0} \\
1 & 0 & \mathbf{0} & \mathbf{0} \\
0 & 1 & \mathbf{0} & \mathbf{0} \\
0 & 1 & \mathbf{0} & \mathbf{0} \\
0 & 1 & \mathbf{0} & \mathbf{0} \\
0 & 1 & \mathbf{0} & \mathbf{0} \\ \hline
0 & 0 & \mathbf{1} & 0 \\
2 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 \\
2 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 \\ \hline
0 & 0 & 0 & \mathbf{1} \\
0 & 2 & 0 & -1 \\
0 & 0 & 0 & 1 \\
0 & 2 & 0 & -1 \\
0 & 0 & 0 & 1 \\
0 & 2 & 0 & -1
\end{array}\right)
\end{array}
\begin{array}{c} \\ \\ \\ \\ \uparrow \\ \\ \\ \text{gap} \\ \\ \\ \downarrow \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}
= \boldsymbol{H}(5).
$$

In the canonical basis we see the appearance of zeros in the top part corresponding to the degrees 0, 1, 2, *etc.* of the columns 3 and 4, as a function of the degree d. The observation that the linear independent monomials are shifted to the high degrees as d increases, is also expressed here. As d increases, a gap between the linear independent monomials corresponding to the affine roots and the linear independent monomials corresponding to the roots at infinity emerges.

We will employ this *mind-the-gap phenomenon* to separate the affine roots and the roots at infinity. In Figure 2.3 we visualize this observation. Although we illustrate it here for the canonical basis of the null space only, the same rank properties hold for the other bases.

We call d_G the degree at which the gap between the two sets of linearly independent monomials occurs. In the example we have $d_G = 4$. The two affine roots can now be computed by using only the first two columns of $\boldsymbol{H}(4)$ to phrase the eigenvalue problem as in the previous examples. In order to ensure that the root-finding problem has no interference from the roots at infinity, the shift relation may not map onto monomials of a degree that is higher than $d_G + 1$. Alternatively, if one wants to use a shift polynomial g with $\deg(g) > 1$, the root-finding procedure should be executed on $\boldsymbol{M}(d)$ and $\boldsymbol{H}(d)$ with $d \geq d_G + \deg(g)$.

For instance, let us consider $g(x_1, x_2) = 2x_1 - 4x_2$. The linearly independent monomials are 1 and x_1. Since the highest degree occurring in the linearly independent monomials is one and the shift polynomial g is of degree two,

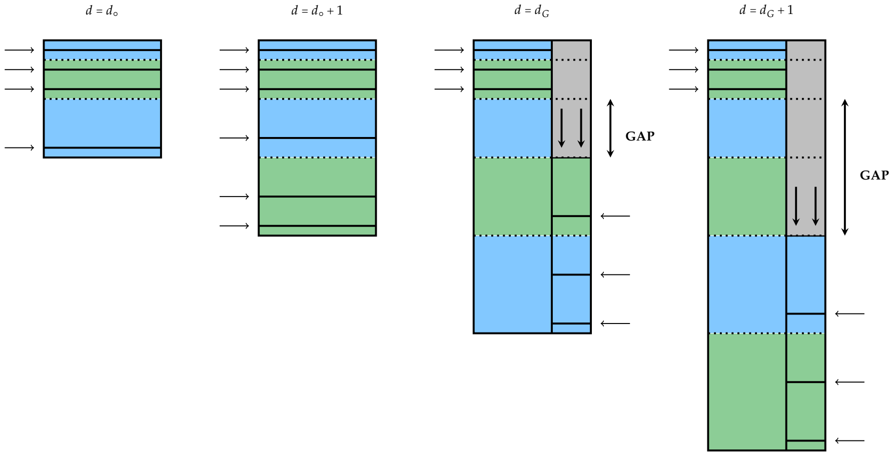

Figure 2.3: Visual representation of the separation of normal set elements of affine roots and solutions at infinity as observed in the canonical basis for the null space $\boldsymbol{H}(d)$. As the degree d increases, the linear independent monomials corresponding to the affine roots stabilize (indicated by the horizontal lines and the arrows on the left-hand-side of the matrix), whereas the linear independent monomials that are caused by the solutions at infinity move along to high degrees (indicated by the horizontal lines in the matrix and the arrows on the right-hand-side of the matrix). Since we are considering the canonical basis for the null space, in the right-most columns the entries on the top are all zero (indicated by the gray block which grows in vertical dimension as d increases), and only entries in the bottom blocks are nonzero. Hence, at a certain degree d a 'gap' emerges which allows us to separate the affine roots and the roots at infinity.

we will need to ensure that the gap in $\boldsymbol{H}(d)$ spans until the degree $d = 1 + 2$. This means that we need to construct $\boldsymbol{M}(5)$ and $\boldsymbol{H}(5)$. After constructing $\boldsymbol{H}(5)$, we retain the top-left block in which we will exploit the shift relation to construct an eigenvalue problem. We have then

$$\boldsymbol{S}_1\boldsymbol{H}_a = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and

$$\boldsymbol{S}_g\boldsymbol{H}_a = \begin{pmatrix} 2 & -4 \\ -4 & 2 \end{pmatrix},$$

where $\boldsymbol{H}_a$ denotes the matrix $\boldsymbol{H}(5)$ of which only the first two columns are retained. The eigenvalue decomposition gives

$$\begin{aligned} \boldsymbol{S}_g\boldsymbol{H}_a &= \boldsymbol{T}\boldsymbol{D}\boldsymbol{T}^{-1} \\ &= \begin{pmatrix} -0.707 & -0.707 \\ -0.707 & 0.707 \end{pmatrix} \begin{pmatrix} -2 & 0 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} -0.707 & -0.707 \\ -0.707 & 0.707 \end{pmatrix}^{-1}. \end{aligned}$$

We reconstruct the multivariate Vandermonde structured null space by computing $\boldsymbol{H}_a\boldsymbol{T}$ and rescaling the columns of the result such that the first row equals ones. We find

$$K_a = \left(\begin{array}{cc} 1 & 1 \\ \hline 1 & -1 \\ 1 & -1 \\ \hline 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ \hline 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ \hline \vdots & \vdots \end{array}\right),$$

from which we correctly retrieve the affine roots as $(1,1)$ and $(-1,-1)$.

2.3.2 Conclusions

The current example provides us with the final ingredients to understand and solve the root-finding problem, namely how to detect and deal with so-called roots at infinity. Below we have summarized the important observations.

1. Algebraic relations between the coefficients and/or zero coefficients may cause so-called roots at infinity. It may happen that we observe variables that are independent to become dependent; however, variables that are dependent, always stay dependent.

2. The so-called 'mind-the-gap phenomenon' emerges in the null space of $\boldsymbol{M}(d)$ as d increases. This allows us to separate affine roots and roots at infinity: the linear independent monomials corresponding to the roots at infinity shift towards higher degrees as the overall degree of the Macaulay matrix increases.

3 Linear Algebra and Realization Theory

In the current chapter, we will briefly highlight a few important notions from linear algebra and realization theory that will constitute the heart of our matrix-based polynomial system solving approach.

First of all, we will discuss how systems of homogeneous linear equations can be solved, and give the geometrical interpretation of this problem using concepts such as rank, column and row space, null space and linear (in)dependence. An important notion here is the complementarity between the indices of the linearly independent/dependent columns of the matrix and the linearly dependent/independent rows of its null space, which will turn out to have great importance in the remainder of this manuscript.

Secondly, we will describe a rudimentary algorithm for computing a canonical basis for the null space of a given (sparse) matrix. This algorithm is important mainly for its didactical purposes, but will be used in Chapter 6 to iteratively compute a basis for the null space of the Macaulay matrix.

Thirdly, we will review the shift invariance property that is prevalent in realization theory and that will also show up in the monomial bases of systems of multivariate polynomial equations. This property is a key property in phrasing the root-finding problem as an eigenvalue problem.

For a more elaborate (basic) introduction to linear algebra and systems theory, we refer the reader to Appendix A.

3.1 Solving Homogeneous Linear Equations

3.1.1 Geometrical Interpretation

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a given matrix, and consider the problem of finding (all) vectors $\boldsymbol{x} \in \boldsymbol{R}^{n \times 1}$ that satisfy $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$.

Let us now interpret this problem 'geometrically' in the column and row space of the matrix $\boldsymbol{A}$. First we discuss the the column space interpretation. We write $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ as

$$\begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} | \\ 0 \\ | \end{pmatrix},$$

with $\boldsymbol{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}^T$, so that we can now write

$$\sum_{i=1}^{n} \boldsymbol{a}_i x_i = \boldsymbol{0}.$$

Provided that $\boldsymbol{x} \neq \boldsymbol{0}$, this means that some columns of $\boldsymbol{A}$ can be written as a linear combination of other columns. The fact that there exist a linear dependency between the columns of $\boldsymbol{A}$ implies that $\boldsymbol{A}$ is not of full column rank.

Next we interpret this problem in the row space of $\boldsymbol{A}$. We write the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ as

$$\begin{pmatrix} - & \boldsymbol{b}_1^T & - \\ - & \boldsymbol{b}_2^T & - \\ & \vdots & \\ - & \boldsymbol{b}_m^T & - \end{pmatrix} \begin{pmatrix} | \\ \boldsymbol{x} \\ | \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Now we have

$$\boldsymbol{b}_j^T \boldsymbol{x} = 0, \qquad \text{with } j = 1, \dots, p.$$

This can be interpreted as follows. We are looking for a vector $\boldsymbol{x}$ that is orthogonal to all rows of the matrix $\boldsymbol{A}$, *i.e.*, the row space of $\boldsymbol{A}$. From the rank-nullity theorem we know that the dimension of the orthogonal complement of the row space is $n - r$-dimensional.

Notice that, in the particular case that $r = n$ (*i.e.*, the matrix $\boldsymbol{A}$ is of full column rank), the only solution to $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{0}$ is $\boldsymbol{X} = \boldsymbol{0}$. Equivalently, the equation $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{0}$ can only have non-trivial solutions, provided that $r < n$. It is important to note that this statement is independent from the number of equations p — in particular, it does not matter whether $p < q$ or $p > q$ or $p = q$; only the column rank of $\boldsymbol{A}$ matters.

3.1.2 Complementarity Columns of A versus Rows of X

The following theorem expresses an interesting complementarity between the rank of the column space of $\boldsymbol{A}$ and the rank of the null space $\boldsymbol{X}$. Later on, this property will be of paramount importance in the interpretation of linearly dependent and linearly independent monomials and how to determine them.

Theorem 3.1. Let us again consider $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\boldsymbol{A}) = r$ and $\boldsymbol{X} \in \mathbb{R}^{n \times n-r}$ such that

$$\boldsymbol{AX} = \boldsymbol{0} \quad \text{with } \text{rank}(\boldsymbol{X}) = n - r.$$

Now reorder the columns of $\boldsymbol{A}$ and then partition them as $\begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{pmatrix}$, where $\boldsymbol{A}_1 \in \mathbb{R}^{m \times n-r}$ and $\boldsymbol{A}_2 \in \mathbb{R}^{m \times r}$, *i.e.*, the block $\boldsymbol{A}_2$ contains r linearly independent columns. This reordering and partitioning is generally not unique, but it can always be done. We partition the rows of $\boldsymbol{X}$ accordingly. Formally this is written as

$$\begin{aligned} \boldsymbol{AX} &= \boldsymbol{0} \\ \boldsymbol{A} \begin{pmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{P}_1^T \\ \boldsymbol{P}_2^T \end{pmatrix} \boldsymbol{X} &= \boldsymbol{0}, \\ \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix} &= \boldsymbol{0}, \\ \boldsymbol{A}_1 \boldsymbol{X}_1 + \boldsymbol{A}_2 \boldsymbol{X}_2 &= \boldsymbol{0}, \end{aligned}$$

where $\boldsymbol{P} = \begin{pmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 \end{pmatrix}$, with $\boldsymbol{PP}^T = \boldsymbol{I}$, denotes the column permutation matrix that performs the reordering and partitioning of $\boldsymbol{A}$ and $\boldsymbol{X}$. We now have that

$$\text{rank}(\boldsymbol{X}_1) = n - r \quad \Leftrightarrow \quad \text{rank}(\boldsymbol{A}_2) = r.$$

Proof. The $\Leftarrow$ part follows from the following. If $\text{rank}(\boldsymbol{A}_2) = r$, then $\boldsymbol{A}_2^T \boldsymbol{A}_2$ is invertible, and hence $\boldsymbol{X}_2 = -\left(\boldsymbol{A}_2^T \boldsymbol{A}_2\right)^{-1} \boldsymbol{A}_2 \boldsymbol{A}_1 \boldsymbol{X}$, from which follows that

$$\text{rank}(\boldsymbol{X}) = n - r = \text{rank} \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix} = \text{rank} \left(\begin{pmatrix} \boldsymbol{I}_{n-r} \\ -\left(\boldsymbol{A}_2^T \boldsymbol{A}_2\right)^{-1} \boldsymbol{A}_2 \boldsymbol{A}_1 \end{pmatrix} \boldsymbol{X}_1 \right).$$

We also have that

$$\text{rank}(\boldsymbol{X}) = \text{rank}(\boldsymbol{X}_1) = n - r.$$

The $\Rightarrow$ part can be proved as follows. Since $\boldsymbol{X}_1$ is square invertible, we have $\boldsymbol{A}_1 = -\boldsymbol{A}_2 \boldsymbol{X}_2 \boldsymbol{X}_1^{-1}$, so that $\begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{pmatrix} = \boldsymbol{A}_2 \begin{pmatrix} -\boldsymbol{X}_2 \boldsymbol{X}_1^{-1} & \boldsymbol{I}_r \end{pmatrix}$, which shows that

$$\text{rank}(\boldsymbol{A}) = \text{rank}(\boldsymbol{A}_2) = r.$$

□

Let us briefly contemplate this important result as it is paramount for the remainder of this manuscript. Said in words, it states that the set of unknowns in a system of homogeneous linear equations can always be partitioned into 'dependent' variables (*i.e.,* $\boldsymbol{X}_2$) and 'independent' variables (*i.e.,* $\boldsymbol{X}_1$), meaning that $\boldsymbol{X}_2$ can be written as a linear combination of $\boldsymbol{X}_1$.

Alternatively, the matrix $\boldsymbol{A}$ can be partitioned into columns that are linearly independent (*i.e.,* the columns of $\boldsymbol{A}_2$) and columns that can be written as linear combinations of the independent ones (*i.e.,* the columns of $\boldsymbol{A}_1$).

A consequence of this result is that, in the null space of a given matrix $\boldsymbol{A}$ of rank r, which is $n-r$-dimensional, one can always find a basis in which each of the basis vectors (columns) has at least $n-r-1$ zeros:

$$\begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{n-r} \\ \boldsymbol{X}_2\boldsymbol{X}_1^{-1} \end{pmatrix} = 0.$$

We call such a basis for the null space a *canonical basis*.

Notice that such a canonical basis is not unique. In general there are $\binom{n}{r}$ combinations to choose r columns out of n columns. Not all of these choices lead to a $m \times r$ sub-matrix of $\boldsymbol{A}$ that is of full column rank r. However, for every valid set of independent variables (*i.e.,* corresponding to a sub-matrix of $\boldsymbol{A}$ that is of rank r), there is a different canonical basis for the null space.

It is important to emphasize the duality property in the above results. The sets of indices of linear independent columns of $\boldsymbol{A}$ and linear independent rows of $\boldsymbol{X}$, are complementary.

Corollary 3.2. For every selection of columns of $\boldsymbol{A}$, collected in a sub-matrix $\boldsymbol{A}_2$ of full column rank r, the sub-matrix of $\boldsymbol{X}$ formed by the 'complementary' selection of rows in $\boldsymbol{X}$ will be of full rank $n-r$.

This also implies that for a selection of r columns of $\boldsymbol{A}$ in a sub-matrix $\boldsymbol{A}_2$ that is *not* of full column rank, the corresponding sub-matrix of $\boldsymbol{X}$ formed from the complementary selection of rows, will *not* be of full rank.

Remark 3.3. In this manuscript, we typically use an ordering on the variables, *i.e.,* the unknowns $\boldsymbol{x} := \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}^T$ in the problem, in which x_1 precedes x_2, x_3, *etc.* Often, we would like to have the set of linear independent variables to have indices that are as small as possible in the particular ordering we are using. This will imply that we are interested in finding the *first* $n-r$ rows of $\boldsymbol{X}$ that are linear independent. The indices of the columns in $\boldsymbol{A}$ that are linearly independent follow from the complement of indices, *i.e.,* they correspond to the r linear independent columns of $\boldsymbol{A}$ that one can find, when starting from the right to the left of $\boldsymbol{A}$.

3.2 Motzkin Null Space Computation

In the current section we will develop a method for computing a canonical basis for the null space of a matrix. Due to possible numerical issues, the method is not suitable to tackle big matrices, but for didactical purposes, we have chosen to include the procedure here. Especially in the case that we are dealing with a sparse matrix, the method is conceptually interesting. Furthermore, the sparse null space computations we will develop in Chapter 6 are inspired on this algorithm.

The Motzkin approach proceeds by constructing null space vectors of a single row by forming pair-wise eliminations on the non-zero entries. By considering one row at a time, and pre-multiplying the row under consideration with the product of all computed bases for the null spaces of the previous rows, a *canonical null space* of a full matrix is constructed. This canonical null space has the interesting property that an identity matrix sits in the linearly independent rows, which can hence be read off from the canonical null space.

3.2.1 Null Space of a Single Row

The core of the Motzkin algorithm can best be described by looking at the action on a single row. Given a row vector $\boldsymbol{b}^T$, the Motzkin procedure generates a matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n-1}$ with $\text{rank}(\boldsymbol{W}) = n-1$ and each column of $\boldsymbol{W}$ is orthogonal to $\boldsymbol{b}^T$, or $\boldsymbol{b}^T \boldsymbol{W} = 0$.

The Motzkin (canonical) null space construction algorithm for one row is described below (Algorithm 1). Note that a pre-condition for this algorithm is that $\boldsymbol{b}^T \neq 0$. In the case that $\boldsymbol{b}^T = 0$, the null space can be trivially determined as $\boldsymbol{W} = \boldsymbol{I}_n$.

Algorithm 1. *Motzkin Null Space Construction Row* (`MotzkinRow`)

```
input:   $\boldsymbol{b}^T \in \mathbb{R}^{1 \times n}$ where $\boldsymbol{b}^T \neq 0$
output:  $\boldsymbol{W} \in \mathbb{R}^{n \times n-1}$, such that $\boldsymbol{b}^T \boldsymbol{W} = 0$ and $\text{rank}(\boldsymbol{W}) = n-1$

  1. Determine (right-most) nonzero pivot $p = \boldsymbol{b}^T(i_p)$, with $\boldsymbol{b}^T(i) = 0$ for $i > i_p$
  2. Rescale $\boldsymbol{b}$ as $\boldsymbol{b}^T = \boldsymbol{b}^T / p$
  3. for $i = i_p + 1, \dots, n$, do
       a) $\boldsymbol{W}(:, i-1) = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ denotes the $i$-th standard basis vector
     done
  4. for $i = i_p - 1, \dots, 1$, do
       a) $\boldsymbol{W}(i, i) = 1$
```

```
    b) $W(i_p, i) = -b^T(i)$
done
```

3.2.2 Motzkin Null Space of a Matrix

One can now repeatedly apply the procedure described in the previous paragraph to construct the canonical basis for the null space of any given matrix A.

The idea behind the `MotzkinMatrix` procedure is as follows. Given a full matrix A, the Motzkin procedure takes the consecutive rows of A, i.e., a_i^T, into account.

Let us start with row a_1^T. Algorithm 1 constructs a matrix W_1 which forms a basis for the null space of a_1^T. In the next step, we consider a_2^T. First, this row vector is converted into $b_2^T = a_2^T W_1$, and the `MotzkinRow` procedure is performed for b_2^T, resulting in W_2. When the third row of A is considered, we first convert this to $b_3^T = a_3^T W_1 W_2$ and then find W_3.

As the row count is increased to k, this procedure guarantees that the product of the matrices W_i for $i = 1, \ldots, k-1$ is composed of vectors that are orthogonal to all previous rows of A, *i.e.*, $a_1^T, a_2^T, \ldots, a_{k-1}^T$. Ultimately, as the last row of A is processed, the complete null space matrix H is computed as $H = W_1 W_2 \ldots W_n$.

The canonical (Motzkin) null space construction algorithm for a complete matrix is described in Algorithm 2. This procedure leads to a very sparse representation of the null space and would use a rather limited amount of memory since the coefficients of b_k occur very predictably in W_k and the multiplication $W_1 W_2 \cdots W_k$ involving very sparse matrices can be implemented efficiently.

However, the Motzkin procedure is numerically flawed: during the consecutive multiplication of the matrices W_k, some elements of $b_k W_1 W_2 \cdots W_{k-1}$ may become very small — choosing one of them as a non-zero pivot elements would lead to an incorrect result.

Algorithm 2. *Motzkin Null Space Construction Matrix* (`MotzkinMatrix`)

input: $A \in \mathbb{R}^{m \times n}$, with $\text{rank}(A) = r$
output: $H \in \mathbb{R}^{n \times n-r}$, such that $AH = 0$ and $\text{rank}(H) = n - r$

```
1. for $i = 1, \ldots, m$, do
    a) if $i = 1$, do
        i. $a_1^T = A(1, :)$
```

```
          ii. $b_1^T = a_1^T$
          iii. $W_1 = \texttt{MotzkinRow}(b_1^T)$
      b) else, do
          i. $a_i^T = A(i,:)$
          ii. $b_i^T = a_i^T W_1 W_2 \dots W_{i-1}$
          iii. $W_i = \texttt{MotzkinRow}(b_i^T)$
         done
    done
  2. $H = W_1 W_2 W_3 \dots W_m$
```

Let us consider a small example in which we compute a basis for the null space of a 3 × 5 matrix $\boldsymbol{A}$.

Example 3.4. Consider the matrix $\boldsymbol{A}$ given as

$$\boldsymbol{A} = \begin{pmatrix} 0 & 2 & 1 & 0 & 2 & 0 \\ 1 & 0 & 3 & 2 & 0 & 1 \\ 4 & 0 & 3 & 2 & 1 & 0 \end{pmatrix}. \tag{3.1}$$

First we compute the Motzkin null space for the first row

$$\boldsymbol{a}_1^T = \begin{pmatrix} 0 & 2 & 1 & 0 & \textcircled{2} & 0 \end{pmatrix},$$

which gives

$$\boldsymbol{W}_1 = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & -1 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} \end{pmatrix}. \tag{3.2}$$

The second row $\boldsymbol{a}_2^T$ is first multiplied with $\boldsymbol{W}_1$ to obtain

$$\boldsymbol{b}_2^T = \boldsymbol{a}_2^T \boldsymbol{W}_1 = \begin{pmatrix} 1 & 0 & 3 & 2 & \textcircled{1} \end{pmatrix}.$$

The Motzkin null space $\boldsymbol{W}_2$ is found as

$$\boldsymbol{W}_2 = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ -1 & 0 & -3 & -2 \end{pmatrix}. \tag{3.3}$$

The third row is converted to

$$\boldsymbol{b}_3^T = \boldsymbol{a}_3^T \boldsymbol{W}_1 \boldsymbol{W}_2 = \begin{pmatrix} 4 & -1 & 2.5 & \textcircled{\textbf{2}} \end{pmatrix},$$

and the Motzkin null space is found as

$$\boldsymbol{W}_3 = \begin{pmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \\ -2 & 0.5 & -1.25 \end{pmatrix}. \tag{3.4}$$

Finally, we find a canonical basis for the null space of $\boldsymbol{A}$ as

$$\boldsymbol{H} = \boldsymbol{W}_1 \boldsymbol{W}_2 \boldsymbol{W}_3 = \begin{pmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \\ -2 & 0.5 & -1.25 \\ 0 & -1 & -0.5 \\ 3 & -1 & -0.5 \end{pmatrix}. \tag{3.5}$$

3.3 Realization Theory Concepts

3.3.1 Realization Theory for 1D Systems

Autonomous Descriptor System

Consider the state equation in the Kronecker canonical form (Appendix A) as

$$\begin{pmatrix} \boldsymbol{v}(k+1) \\ \hline \boldsymbol{w}(k-1) \end{pmatrix} = \left(\begin{array}{c|c} \boldsymbol{A} & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{E} \end{array}\right) \begin{pmatrix} \boldsymbol{v}(k) \\ \hline \boldsymbol{w}(k) \end{pmatrix},$$

with $\boldsymbol{v} \in \mathbb{R}^{\theta_R}$, $\boldsymbol{w} \in \mathbb{R}^{\theta_S}$, $\boldsymbol{A} \in \mathbb{R}^{\theta_R \times \theta_R}$, and $\boldsymbol{E} \in \mathbb{R}^{\theta_S \times \theta_S}$. The initial states are given by $\boldsymbol{v}_0 := \boldsymbol{v}(0)$ and $\boldsymbol{w}_d := \boldsymbol{w}(d)$.

By iterating the state equations we find the so-called state sequence matrices

$$\begin{aligned} \boldsymbol{V}_{0|d} &:= \begin{pmatrix} | & | & | & & | \\ \boldsymbol{v}(0) & \boldsymbol{v}(1) & \boldsymbol{v}(2) & \cdots & \boldsymbol{v}(d) \\ | & | & | & & | \end{pmatrix}, \\ &= \begin{pmatrix} | & | & | & & | \\ \boldsymbol{v}(0) & \boldsymbol{A}\boldsymbol{v}(0) & \boldsymbol{A}^2\boldsymbol{v}(0) & \cdots & \boldsymbol{A}^d\boldsymbol{v}(0) \\ | & | & | & & | \end{pmatrix}, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{W}_{0|d} &:= \begin{pmatrix} | & | & | & & | \\ \boldsymbol{w}(0) & \boldsymbol{w}(1) & \boldsymbol{w}(2) & \cdots & \boldsymbol{w}(d) \\ | & | & | & & | \end{pmatrix}, \\ &= \begin{pmatrix} | & | & | & & | \\ \boldsymbol{E}^{d}\boldsymbol{w}(d) & \boldsymbol{E}^{d-1}\boldsymbol{w}(d) & \boldsymbol{E}^{d-2}\boldsymbol{w}(d) & \cdots & \boldsymbol{w}(d) \\ | & | & | & & | \end{pmatrix}. \end{aligned}$$

Shift Invariance

The state sequence matrices $\boldsymbol{V}_{0|d}$ and $\boldsymbol{W}_{0|d}$ exhibit a Vandermonde-like shift structure. We have the following:

$$\boldsymbol{V}_{0|d-1}^T \boldsymbol{A}^T = \boldsymbol{V}_{1|d}^T,$$

and

$$\boldsymbol{W}_{1|d}^T \boldsymbol{E}^T = \boldsymbol{W}_{0|d-1}^T,$$

from which $\boldsymbol{A}$ and $\boldsymbol{E}$ can be determined.

Exploiting this shift invariance is a central tool in realization theory (see Appendix A) and will turn out to be essential in the linear algebra framework for solving systems of polynomial equations.

Example 3.5. Suppose that the matrix $\boldsymbol{V}_{0|4}$ is given as

$$\boldsymbol{V}_{0|4} = \begin{pmatrix} 1 & 0 & -6 & 30 & -114 \\ 0 & 1 & 5 & 19 & 65 \end{pmatrix}.$$

We have now

$$\boldsymbol{A}^T = (\boldsymbol{V}_{0|3}^T)^{+} \boldsymbol{V}_{1|4}^T = \begin{pmatrix} 0 & 1 \\ -6 & 5 \end{pmatrix},$$

which could be read off from $\boldsymbol{V}_{0|4}$ because the first two columns are the identity matrix.

3.3.2 Realization Theory for nD Systems

Several nD state space descriptions have been developed, see *e.g.,* Attasi (1976); Gałkowski (2001). For the problem of solving a system of polynomial equations, the model by Attasi (1976) is a natural starting point as illustrated in Hanzon and Hazewinkel (2006).

We employ a simplified version of Attasi (1976), defined as follows:

$$v(k_1, \ldots, k_{i-1}, k_i + 1, k_{i+1}, \ldots, k_n) = \boldsymbol{A}_i v(k_1, \ldots, k_n),$$

for all $i = 1, \ldots, n$. We have $v \in \mathbb{R}^{\theta}$. The action matrices $A_i \in \mathbb{R}^{\theta \times \theta}$ form a commuting family of matrices: we have that $A_i A_j = A_j A_i$, for all $i, j \in \{1, \ldots, n\}$.

Iterating the state equations leads to a multivariate generalization of the Vandermonde structure observed in the 1D case. The multivariate Vandermonde structure can again be used to determine the action matrices A_i.

Let us illustrate these concepts by means of a simple example where we take $n = 2$ and $d = 3$.

Example 3.6. Let $n = 2$ and $d = 3$. The state sequence matrix $V_{0|d}$ is found as[1]

$$V_{0|3} = \left(\begin{array}{c|cc|ccc|cccc} | & | & | & | & | & | & | & | & | & | \\ v_{00} & v_{10} & v_{01} & v_{20} & v_{11} & v_{02} & v_{30} & v_{21} & v_{12} & v_{03} \\ | & | & | & | & | & | & | & | & | & | \end{array}\right),$$

$$= \left(\begin{array}{c|cc|c|cccc} | & | & | & & | & | & | & | \\ v_{00} & A_1 v_{00} & A_2 v_{00} & \cdots & A_1^3 v_{00} & A_1^2 A_2 v_{00} & A_1 A_2^2 v_{00} & A_2^3 v_{00} \\ | & | & | & & | & | & | & | \end{array}\right).$$

Note that the order in which the states are iterated is not uniquely determined. Here we have used an ordering that is compatible to the degree negative lexicographic ordering (Definition 5.1), but other orderings can be used as well.

The multivariate Vandermonde shift structure leads to expressions of the form

$$\left(\begin{array}{c} v_{00} \\ \hline v_{10} \\ v_{01} \\ \hline v_{20} \\ v_{11} \\ v_{02} \end{array}\right) A_1^T = \left(\begin{array}{c} v_{10} \\ \hline v_{20} \\ v_{11} \\ \hline v_{30} \\ v_{21} \\ v_{12} \end{array}\right),$$

and

$$\left(\begin{array}{c} v_{00} \\ \hline v_{10} \\ v_{01} \\ \hline v_{20} \\ v_{11} \\ v_{02} \end{array}\right) A_2^T = \left(\begin{array}{c} v_{01} \\ \hline v_{11} \\ v_{02} \\ \hline v_{21} \\ v_{12} \\ v_{03} \end{array}\right).$$

[1] We have employed a simplified notation $v_{kl} := v(k, l)$ to avoid an overloaded notation.

Part II

Polynomial System Solving

Sylvester Matrix Formulation 4

In this chapter, we will study in detail the problem of finding the common roots of a system of univariate polynomial equations. The univariate approach provides us with the main ingredients to tackle the multivariate case.

Apart from the well-known fact that univariate root-finding is intimately linked to matrix eigenvalue problems, we will review a result due to Sylvester that asserts whether two univariate polynomials have a root in common. A natural consequence is then that the common root(s) themselves can be computed as well. This will lead to the formulation of an eigenvalue problem from which all the common roots can be computed. Finally we will highlight a natural link between the Sylvester root-finding approach and realization theory that is largely unknown in the literature.

4.1 Univariate Root-finding

4.1.1 Companion Matrix

One of the first facts in studying linear algebra and eigenvalue problems is that the eigenvalues of a $n \times n$ matrix $\boldsymbol{A}$ are given by the roots of its characteristic polynomial $p(\lambda)$, which is defined as

$$p(\lambda) = \det(\boldsymbol{A} - \lambda \boldsymbol{I}_n). \tag{4.1}$$

The converse step can be taken as well: with every univariate polynomial $p(x)$, a matrix can be associated of which the eigenvalues correspond to the roots of p. Such a matrix is called a companion matrix, and the best known formulation is the Frobenius companion matrix, which we will review here.

Proposition 4.1. Consider a polynomial

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0. \tag{4.2}$$

The following expression shows that the roots of $p(x)$, *i.e.,* the values x for which $p(x) = 0$, correspond to the eigenvalues of the so-called Frobenius companion matrix as in (1.1):

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & & & 1 & 0 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-2} & -a_{n-1} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-2} \\ x^{n-1} \end{pmatrix} x.$$

This matrix is the basis for many numerical root-finding methods. For example, the `roots` command in MATLAB computes the roots of a polynomial equation by finding the eigenvalues of the Frobenius companion matrix.

4.1.2 Finding the Common Roots of a System of Two Univariate Polynomials

In Chapter 1 we have reviewed the fact that the eigenvalues of the Frobenius companion matrix correspond to the roots of a univariate polynomial. We will now discuss a related concept which employs linear algebra to finding the common roots of univariate polynomials.

The well-known construction by Sylvester (1853) tests whether two polynomials $f_1(x)$ and $f_2(x)$ have common roots by investigating the determinant of a structured square matrix built from the coefficients of the equations. Consider the system

$$\begin{cases} f_1(x) & = & a_r x^r + a_{r-1} x^{r-1} + \ldots + a_0 & = & 0, \\ f_2(x) & = & b_s x^s + b_{s-1} x^{s-1} + \ldots + b_0 & = & 0, \end{cases}$$

having m single common roots (*i.e.,* no common roots with multiplicity).

Definition 4.2 (Sylvester Matrix for Two Equations). Multiplying $f_1(x)$ and $f_2(x)$ with powers of x gives rise to a square system of linear equations $\boldsymbol{M}\,\boldsymbol{k} =$

0, or

$$\begin{array}{r} s \text{ rows} \left\{ \begin{array}{c} \\ \\ \\ \\ \end{array} \right. \\ r \text{ rows} \left\{ \begin{array}{c} \\ \\ \\ \\ \end{array} \right. \end{array} \left(\begin{array}{ccccccc} a_0 & a_1 & \dots & a_r & & & \\ & a_0 & a_1 & \dots & a_r & & \\ & & \ddots & \ddots & & \ddots & \\ & & & a_0 & a_1 & \dots & a_r \\ \hline b_0 & b_1 & \dots & b_s & & & \\ & b_0 & b_1 & \dots & b_s & & \\ & & \ddots & \ddots & & \ddots & \\ & & & b_0 & b_1 & \dots & b_s \end{array} \right) \left(\begin{array}{c} x^0 \\ x^1 \\ \vdots \\ x^{r+s-1} \end{array} \right) = \left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array} \right).$$

The coefficient matrix $\boldsymbol{M}$ is called the *Sylvester matrix*, and has size $(r+s) \times (r+s)$. The f_1-block of the Sylvester matrix has s rows, and the f_2-block has r rows. The monomial basis vector $\boldsymbol{k}$ contains as its elements the monomials x^d for $d = 0, \dots, r+s-1$ and has a Vandermonde structure.

Proposition 4.3 (Cox et al. (2005, 2007))**.** The Sylvester matrix $\boldsymbol{M}$ has a zero determinant if its composing polynomials $f_1(x)$ and $f_2(x)$ have a common root.

Indeed, evaluating $\boldsymbol{k} = \begin{pmatrix} 1 & x & x^2 & \dots & x^{r+s-1} \end{pmatrix}^T$ at a common root $x^{(i)}$ of $f_1(x)$ and $f_2(x)$ gives rise to a non-zero vector in the null space of the Sylvester matrix $\boldsymbol{M}$.

Interestingly, Sylvester's construction can be employed to determine the common roots. An important tool in understanding how this can be achieved is the Vandermonde basis of the Sylvester matrix composed of the (for the moment unknown) evaluations of the common roots in the Vandermonde monomial basis vector $\boldsymbol{k}$.

Definition 4.4 (Univariate Vandermonde Null Space)**.** Assume that the polynomials $f_1(x)$ and $f_2(x)$ have m single common roots $x^{(i)}$, for $i = 1, \dots, m$, with $x^{(i)} \neq x^{(j)}$, for $j \neq i$. Let $\boldsymbol{K}$ be the matrix that contains as its columns the Vandermonde monomial basis vectors $\boldsymbol{k}$ evaluated at the m common roots $x^{(i)}$,

$$\boldsymbol{K} := \begin{pmatrix} | & | & | \\ \cdots & \boldsymbol{k}|_{x^{(i)}} & \cdots \\ | & | & | \end{pmatrix},$$

which is called the Vandermonde null space of $\boldsymbol{M}$.

The next ingredient is the observation that a multiplication property holds in the Vandermonde monomial vector $\boldsymbol{k}$, *e.g.*, one has

$$\begin{pmatrix} 1 & x & x^2 & \dots & x^{r+s-2} \end{pmatrix}^T x = \begin{pmatrix} x & x^2 & x^3 & \dots & x^{r+s-1} \end{pmatrix}^T$$

which can be represented as $\boldsymbol{S}_1 \boldsymbol{k} x = \boldsymbol{S}_x \boldsymbol{k}$, where $\boldsymbol{S}_1$ is a row selection matrix selecting the rows 1 up to $r+s-1$, and $\boldsymbol{S}_x$ is a row selection matrix selecting the rows 2 up to $r+s$. This relation also holds for the monomial basis vectors evaluated at each of the m roots, when multiplying with the i-th root $x^{(i)}$, for $i = 1, \ldots, m$, *i.e.*, $\boldsymbol{S}_1 \, \boldsymbol{k}|_{x^{(i)}} \, x^{(i)} = \boldsymbol{S}_1 \, \boldsymbol{k}|_{x^{(i)}}$. Applying the multiplication property to all m roots gives

$$\boldsymbol{S}_1 \boldsymbol{K} \boldsymbol{D}_x = \boldsymbol{S}_x \boldsymbol{K}, \tag{4.3}$$

where $\boldsymbol{K}$ represents the multivariate Vandermonde null space of $\boldsymbol{M}$, and $\boldsymbol{D}_x$ is a diagonal matrix containing the m roots, *i.e.*, $\boldsymbol{D}_x = \operatorname{diag}\left(x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right)$. Unfortunately the null space of $\boldsymbol{M}$ with the canonical structure $\boldsymbol{K}$ is not directly available. Instead a basis for the null space of $\boldsymbol{M}$ can be computed as $\boldsymbol{Z}$, which is related to the multivariate Vandermonde basis for the null space by $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$, with $\boldsymbol{T}$ non-singular. Combining the previous observations easily leads to an eigenvalue problem from which the common roots can be obtained.

Theorem 4.5 (Sylvester matrix univariate root-finding)**.** Together with (4.3) the problem of finding the common roots of f_1 and f_2 is reduced to the eigendecomposition $\boldsymbol{T}\boldsymbol{D}_x\boldsymbol{T}^{-1} = \left(\boldsymbol{S}_1\boldsymbol{Z}\right)^{+} \boldsymbol{S}_x\boldsymbol{Z}$, with $(\cdot)^{+}$ denoting the Moore-Penrose pseudo-inverse. The eigenvalues of $\left(\boldsymbol{S}_1\boldsymbol{Z}\right)^{+} \boldsymbol{S}_x\boldsymbol{Z}$ are the m common roots of $f_1(x)$ and $f_2(x)$.

From the computation of $\boldsymbol{Z}\boldsymbol{T}$ and consequently normalizing the result column-wise such that the first row consists of ones, a reconstruction of the Vandermonde null space exhibiting the Vandermonde structure is obtained. All solutions (and their powers) can now be read off directly.

Example 4.6. Consider the polynomials

$$\begin{array}{rclcl} f_1(x) & = & (x-1)(x-2) & = & x^2 - 3x + 2, \\ f_2(x) & = & (x-1)(x-2)(x+1) & = & x^3 - 2x^2 - x + 2, \end{array}$$

having two common roots $x^{(1)} = 1$ and $x^{(2)} = 2$. The Sylvester matrix $\boldsymbol{M}$ is constructed as

$$\boldsymbol{M} = \begin{array}{c} \\ f_1 \\ xf_1 \\ x^2 f_1 \\ f_2 \\ xf_2 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & x & x^2 & x^3 & x^4 \end{array} \\ \begin{pmatrix} 2 & -3 & 1 & 0 & 0 \\ 0 & 2 & -3 & 1 & 0 \\ 0 & 0 & 2 & -3 & 1 \\ 2 & -1 & -2 & 1 & 0 \\ 0 & 2 & -1 & -2 & 1 \end{pmatrix} \end{array}.$$

The null space of $\boldsymbol{M}$ has a dimension of two (the singular values are 5.4434, 4.8990, 2.8931, 0.0000 and 0.0000), and a numerical basis $\boldsymbol{Z}$ for the null space is computed using a singular value decomposition. The eigenvalue decomposition of $\left(\boldsymbol{S}_1\boldsymbol{Z}\right)^{+} \boldsymbol{S}_x\boldsymbol{Z}$ reveals the common roots of $f_1(x)$ and $f_2(x)$ as $x^{(1)} = 1$ and $x^{(2)} = 2$ (exact up to machine precision).

4.1.3 Systems with More Than Two Equations

Although it is classically defined for two equations, the Sylvester construction can be performed for any number s of polynomials $f_i(x)$, with $i = 1, \ldots, s$.

Definition 4.7 (Sylvester Matrix)**.** For a set of s equations $f_1, f_2, \ldots, f_s$ the Sylvester matrix $\boldsymbol{M}(d)$ for degree d is constructed as above: There are s blocks, each of which contains the coefficients of a single equation $f_i = 0$, with $i = 1, \ldots, s$. The columns of $\boldsymbol{M}(d)$ are indexed by the monomials $1, x, x^2, \ldots, x_d$. The Vandermonde monomial basis $\boldsymbol{k}(d)$ contains the monomials $1, x, x^2, \ldots, x_d$.

To determine the number of common roots, one inspects the nullity of the Sylvester matrix $\boldsymbol{M}(d)$ as d increases: we have seen that every common root defines a vector in the null space of $\boldsymbol{M}$. After the nullity of the Sylvester matrix has stabilized to a constant, Theorem 4.5 can be used on $\boldsymbol{Z}(d)$ to retrieve the m common roots (Serpedin and Giannakis, 1999). We call $d^\star$ the degree at which the nullity stabilizes.

Let us now have a look at an example, where the Sylvester matrix of three equations is constructed.

Example 4.8. Consider the equations

$$\begin{array}{rclcrcl} f_1(x) &=& x^2(x-1)(x-2) &=& x^5 - 7x^3 + 6x^2 &=& 0, \\ f_2(x) &=& x^3(x-1)(x-2) &=& x^5 - 3x^4 + 2x^3 &=& 0, \\ f_3(x) &=& (x+1)^3(x-1)(x-2) &=& x^5 - 4x^3 - 2x^2 + 3x + 2 &=& 0, \end{array}$$

exhibiting the common roots $x = 1$ and $x = 2$.

Because the dimensions of the Sylvester matrix rapidly become too large to print, we have summarized the most important properties in Table 4.1.

Table 4.1: Diagram showing the properties of the Sylvester matrix $\boldsymbol{M}(d)$ as a function of the degree d. The nullity of the Sylvester matrix stabilizes at the value $m = 2$ at degree $d = 7$.

d	size $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	roots (eigenvalues)	correct?
5	3×6	3	$1, 2, -0.4658$	×
6	6×7	2	$1, 2$	✓
7	9×8	2	$1, 2$	✓
8	12×9	2	$1, 2$	✓

4.1.4 Multiple Roots and Differential Operators

A single univariate polynomial $f(x)$ may have multiple roots, consider for instance the equation

$$f(x) = (x-3)^2(x-1)^3(x+2) = 0,$$

which has a double root at $x = 3$, a triple root at $x = 1$ and a single root at $x = -2$. Formally, we say that $f(x)$ has an μ-fold root $x^\star$ iff $\frac{d^k f}{dx^k}(x^\star) = 0$, for $k = 0, 1, \ldots, \mu - 1$ and $\frac{d^\mu f}{dx^\mu}(x^\star) \neq 0$.

A system of univariate polynomials may also have common roots with multiplicity. In terms of the Sylvester matrix and the Vandermonde basis vector, this translates into the fact that for the root in consideration only a single Vandermonde structured basis vector, evaluated at the common root, can be constructed that lies in the null space of the Sylvester matrix. As in the single equation case, we also need to take into account differentials: the differentials of the Vandermonde vector, evaluated at the common root, are also elements of the null space. This leads to the so-called *generalized Vandermonde matrix* (Serpedin and Giannakis, 1999).

Definition 4.9 (Generalized Vandermonde vectors)**.** Let the Sylvester matrix of a given system be denoted by $\boldsymbol{M}$. For every common root $x^{(i)}, i = 1, \ldots, l$ with multiplicity μ_i, where $\sum_i^l \mu_i = m =:$ nullity $\boldsymbol{M}$, the Vandermonde structured basis for the null space has as its elements vectors of the form

$$\frac{1}{(\alpha_i)!} \frac{\partial^{\alpha_i} \boldsymbol{k}}{\partial x^{\alpha_i}}\bigg|_{x^{(i)}},$$

where $\alpha_i = 0, 1, \ldots, \mu_i - 1$.

Example 4.10. Consider the system

$$\begin{array}{lclclcl} f_1(x) & = & (x-2)^3(x+3) & = & x^4 - 3x^3 - 6x^2 + 28x - 24 & = & 0, \\ f_2(x) & = & (x-2)^3(x+3)(x+1) & = & x^5 - 2x^4 - 9x^3 + 22x^2 + 4x - 24 & = & 0, \end{array}$$

having a triple root $x^{(1)} = 2$ and a single root $x^{(2)} = -3$. Denote by $\boldsymbol{M} := \boldsymbol{M}(8)$ the Sylvester matrix built from the polynomials (we have that $r + s - 1 = 4 + 5 - 1 = 8$). The generalized Vandermonde matrix for degree $d = 8$ and the given roots with their multiplicities is constructed as

$$\boldsymbol{K} = \begin{pmatrix} | & | & | & | \\ \boldsymbol{k}|_{x^{(1)}} & \frac{\partial \boldsymbol{k}}{\partial x}\Big|_{x^{(1)}} & \frac{1}{2}\frac{\partial^2 \boldsymbol{k}}{\partial x^2}\Big|_{x^{(1)}} & \boldsymbol{k}|_{x^{(2)}} \\ | & | & | & | \end{pmatrix},$$

or

$$\boldsymbol{K} = \left(\begin{array}{rrr|r} 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & -3 \\ 4 & 4 & 1 & 9 \\ 8 & 12 & 6 & -27 \\ 16 & 32 & 24 & 81 \\ 32 & 80 & 80 & -243 \\ 64 & 192 & 240 & 729 \\ 128 & 448 & 672 & -2187 \\ 256 & 1024 & 1792 & 6561 \end{array} \right),$$

where the bar ($|$) denotes the separation between the root $x^{(1)}$ and the root $x^{(2)}$. It can easily be verified that $\boldsymbol{K}$ is of full column rank and lies in the null space of the Sylvester matrix $\boldsymbol{M}$, hence it is a basis for the null space of $\boldsymbol{M}$.

Notice that the Jordan canonical form (Appendix A) naturally arises in the Sylvester formulation with common roots with multiplicity. Let us write the multiplication property $\boldsymbol{S}_1\boldsymbol{K}\boldsymbol{D}_x = \boldsymbol{S}_x\boldsymbol{K}$, where we let $\boldsymbol{S}_1$ select the rows corresponding to the monomials 0 up to $x^{d^\star-1}$. We have

$$\left(\begin{array}{rrr|r} 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & -3 \\ 4 & 4 & 1 & 9 \\ 8 & 12 & 6 & -27 \\ 16 & 32 & 24 & 81 \\ 32 & 80 & 80 & -243 \\ 64 & 192 & 240 & 729 \\ 128 & 448 & 672 & -2187 \end{array} \right) \left(\begin{array}{rrr|r} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 0 \\ \hline 0 & 0 & 0 & -3 \end{array} \right) = \left(\begin{array}{rrr|r} 2 & 1 & 0 & -3 \\ 4 & 4 & 1 & 9 \\ 8 & 12 & 6 & -27 \\ 16 & 32 & 24 & 81 \\ 32 & 80 & 80 & -243 \\ 64 & 192 & 240 & 729 \\ 128 & 448 & 672 & -2187 \\ 256 & 1024 & 1792 & 6561 \end{array} \right).$$

In the matrix $\boldsymbol{D}_x$ we recognize two Jordan blocks (indicated by the horizontal and vertical bars), each of which corresponds to one of the common roots.

Finally, it must be noted that the differentiation operator exhibits an interesting duality property: instead of having the differentiation operators acting on the Vandermonde vectors $\boldsymbol{k}$, they can alternatively be moved to the equations, such that the system of input equations is adjoined with equations obtained by differentiating the input equations. This will lead to removal of the multiplicities of the roots. Although this approach is conceptually elegant, its practical relevance is limited:

- First of all, it is important to note that this procedure only works for a single root for which the multiplicity structure is considered. Indeed, in general, the equations that are obtained by differentiation do not hold for other common roots.

- Second, it is only possible to adjoin the necessary differentiations when the multiplicity structure (of a certain root) is already known, which is in practice not the case when one is given a system of polynomial equations.

Example 4.11. Let us revisit the equations of Example 4.10. The root $x^{(1)}$ had multiplicity 3, so we now adjoin to the equations also the first and second derivatives of the equations, so that we obtain

$$\begin{array}{rclcl} f_1(x) & = & x^4 - 3x^3 - 6x^2 + 28x - 24 & = & 0, \\ (\partial f_1/\partial x)(x) & = & 4x^3 - 9x^2 - 12x + 28 & = & 0, \\ (\partial^2 f_1/\partial x^2)(x) & = & 12x^2 - 18x - 12 & = & 0, \\ f_2(x) & = & x^5 - 2x^4 - 9x^3 + 22x^2 + 4x - 24 & = & 0, \\ (\partial f_2/\partial x)(x) & = & 5x^4 - 8x^3 - 27x^2 + 44x + 4 & = & 0, \\ (\partial^2 f_2/\partial x^2)(x) & = & 20x^3 - 24x^2 - 54x + 44 & = & 0. \end{array}$$

For this system, we find $d^\star = 6$ and we see that the Sylvester matrix $\boldsymbol{M}(6)$ has only one vector in its null space:

$$\boldsymbol{K} = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 8 \\ 16 \\ 32 \\ 64 \end{pmatrix}.$$

4.2 Sylvester and 1D Realization Theory

From linear system theory we know that the roots of the characteristic polynomial play a crucial role in describing and understanding the sequences which satisfy the corresponding difference equation. This is usually described by means of the Z-transform (Kailath, 1998). The link between univariate polynomials and linear dynamical systems hence arises naturally when considering the associated difference equations and their characteristic polynomials. For example, with a difference equation $v(k+2) - 3v(k+1) + 2v(k) = 0$, we can naturally associate the polynomial equation $x^2 - 3x + 2 = 0$.

In this vein, we will show in the current section how the univariate root-finding problem can be interpreted as the application of realization theory to the null space of the Sylvester matrix. Also the case of roots at infinity will be touched, requiring tools from realization theory for 1D descriptor systems. Although in the context of difference equations roots at infinity may seem irrelevant, this exploration will provide us with the necessary ingredients for the extension to the multivariate case, where roots at infinity do often occur.

4.2.1 Interpretation as System of Difference Equations

The interpretation of a polynomial as a time-shift operator acting on a time series is well known, see *e.g.*, Kailath (1998). For the current exposition, we will limit ourselves to the simplest instance of this principle. We have the following definition.

Definition 4.12 (Polynomials as time-shifts)**.** With the monomial x^{α} the shift operator σ^{α} is associated, which acts on a time signal $v(k)$ as

$$\sigma^{\alpha} : v(k) \mapsto v(k+\alpha).$$

Any polynomial equation $p(x) = a_0 + a_1 x + \ldots + a_n x^n = 0$ can hence be associated to the difference equation

$$a_0 v(k) + a_1 v(k+1) + \ldots + a_n v(k+n) = 0.$$

Proposition 4.13 (Annihilator of Sylvester Matrix)**.** The Sylvester matrix $\boldsymbol{M}(d)$ of the equations f_1 and f_2 describing only affine roots is annihilated by a Vandermonde structured basis defined as

$$\boldsymbol{\Gamma}(d) := \begin{pmatrix} - \boldsymbol{v}_0^T \boldsymbol{A}^0 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^1 - \\ \vdots \\ - \boldsymbol{v}_0^T \boldsymbol{A}^d - \end{pmatrix}.$$

This result may come as no surprise. Indeed, from the Cayley-Hamilton theorem it is known that a matrix satisfies its own characteristic equation. The Sylvester matrix defines the characteristic equation of the LTI system as defined by the polynomials f_1 and f_2.

Proposition 4.14. From the Vandermonde structured basis $\boldsymbol{\Gamma}(d)$ we can extract A by using the shift-invariance. Let $\overline{\boldsymbol{\Gamma}}(d)$ denote the matrix $\boldsymbol{\Gamma}(d)$ with the first row removed, and let $\underline{\boldsymbol{\Gamma}}(d)$ denote the matrix $\boldsymbol{\Gamma}(d)$ with the last row removed. Then we have that

$$\underline{\boldsymbol{\Gamma}}(d)\boldsymbol{A} = \overline{\boldsymbol{\Gamma}}(d),$$

or, when the $\overline{(\cdot)}$ and $\underline{(\cdot)}$ operators are expressed by means of row-selection matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_x$,

$$\boldsymbol{S}_1 \boldsymbol{\Gamma} \boldsymbol{A} = \boldsymbol{S}_x \boldsymbol{\Gamma}.$$

4.2.2 Bases for the Null Space of the Sylvester Matrix

We can consider several matrices that annihilate the Sylvester matrix. Let us define the following matrices that have an interesting interpretation as a basis for the null space of the Sylvester matrix.

Definition 4.15 (Univariate Vandermonde Null Space). Assume that $f_1(x)$ and $f_2(x)$ have m single common roots $x^{(i)}$, for $i = 1, \ldots, m$, with $x^{(i)} \neq x^{(j)}$, for $j \neq i$. Let $\boldsymbol{K}$ be the $r + s \times m$ matrix which contains as its columns the vectors $\boldsymbol{k}$ evaluated at the m common roots $x^{(i)}$,

$$\boldsymbol{K} := \begin{pmatrix} | & | & | \\ \cdots & \boldsymbol{k}|_{x^{(i)}} & \cdots \\ | & | & | \end{pmatrix},$$

which is called the Vandermonde null space of $\boldsymbol{M}$.

Definition 4.16 (Numerical Null Space). From the SVD of $\boldsymbol{M}$, we have

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \\ & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{W}_1^T \\ \boldsymbol{W}_2^T \end{pmatrix},$$

where $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \ldots \geq \sigma_r > 0$, and hence $\operatorname{rank}(\boldsymbol{M}) = r$. Then $\boldsymbol{Z} := \boldsymbol{W}_2$ is a numerical basis for the null space of $\boldsymbol{M}$.

Definition 4.17 (Canonical Null Space). The canonical basis for the null space of $\boldsymbol{M}$ is obtained as the result of the Motzkin algorithm of Chapter 3.

The canonical basis $\boldsymbol{H}$ can also be obtained by multiplying $\boldsymbol{Z}$ on the right by the inverse of the matrix composed of the first two linearly independent rows of $\boldsymbol{Z}$. A trivial but interesting property of $\boldsymbol{H}$ is that the first nonzero entry of each column is the element '1'. The corresponding row is a linear independent row.

Let us illustrate the above definitions by a simple example.

Example 4.18. Consider the simple set of two univariate equations

$$\begin{array}{rcccl} (x-2)(x-3) & = & x^2 - 5x + 6 & = & 0, \\ (x-2)(x-3)(x-1) & = & x^3 - 6x^2 + 11x - 6 & = & 0. \end{array}$$

The Sylvester matrix for degree $d = 4$ is the 5×5 matrix $\boldsymbol{M}$

$$\boldsymbol{M} = \begin{pmatrix} 6 & -5 & 1 & & \\ & 6 & -5 & 1 & \\ & & 6 & -5 & 1 \\ -6 & 11 & -6 & 1 & \\ & -6 & 11 & -6 & 1 \end{pmatrix},$$

where the empty spaces represent zero elements. A basis for the null space of $\boldsymbol{M}$ is computed as $\boldsymbol{Z}$ using a singular value decomposition as follows,

$$\boldsymbol{Z} = \begin{pmatrix} -0.2129 & -0.0361 \\ -0.3673 & -0.0772 \\ -0.5591 & -0.1695 \\ -0.5917 & -0.3841 \\ 0.3963 & -0.9036 \end{pmatrix}.$$

Next $\boldsymbol{Z}$ is multiplied on the right with the inverse of the matrix composed of the first two linear independent rows, as to obtain the canonical null space $\boldsymbol{H}$

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -6 & 5 \\ -30 & 19 \\ -114 & 65 \end{pmatrix}.$$

We identify rows 1 and 2 as the linearly independent rows.

Let us now turn to the root-finding question. We expect the null space to have the following structure

$$\boldsymbol{\Gamma} = \begin{pmatrix} - \boldsymbol{v}_0^T \boldsymbol{A}^0 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^1 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^2 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^3 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^4 - \end{pmatrix}.$$

The following shift relation determines the matrix $\boldsymbol{A}$ (up to a state similarity transformation)

$$\boldsymbol{S}_1 \boldsymbol{\Gamma} \boldsymbol{A} = \boldsymbol{S}_x \boldsymbol{\Gamma}$$

or

$$\begin{pmatrix} - \boldsymbol{v}_0^T \boldsymbol{A}^0 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^1 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^2 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^3 - \end{pmatrix} \boldsymbol{A} = \begin{pmatrix} - \boldsymbol{v}_0^T \boldsymbol{A}^1 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^2 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^3 - \\ - \boldsymbol{v}_0^T \boldsymbol{A}^4 - \end{pmatrix}$$

Hence, when using the canonical basis $\boldsymbol{H}$ for the null space of the Sylvester matrix, we have

$$\boldsymbol{A} = (\boldsymbol{S}_1 \boldsymbol{H})^+ \boldsymbol{S}_x \boldsymbol{H}.$$

An eigenvalue decomposition of $\boldsymbol{A}$ reveals the two common solutions $x = 2$ and $x = 3$

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 \\ -6 & 5 \end{pmatrix} = \boldsymbol{V} \boldsymbol{D} \boldsymbol{V}^{-1},$$

with

$$\boldsymbol{V} = \begin{pmatrix} -0.4472 & -0.3162 \\ -0.8944 & -0.9487 \end{pmatrix}, \quad \boldsymbol{D} = \begin{pmatrix} 2 & \\ & 3 \end{pmatrix}$$

Observe now that this problem has a natural interpretation as an 1D system realization as follows.

$$\boldsymbol{v}(k+1) = \begin{pmatrix} 0 & 1 \\ -6 & 5 \end{pmatrix}^T \boldsymbol{v}(k),$$

and the initial state can be found from considering the matrix $\boldsymbol{H}$. Indeed, we have that the first row of $\boldsymbol{H}$ corresponds to $\boldsymbol{v}_0^T\boldsymbol{A}^0 = \boldsymbol{v}_0^T$, or

$$\boldsymbol{v}(0) = \begin{pmatrix} 1 & 0 \end{pmatrix}^T.$$

The Vandermonde null space of $\boldsymbol{M}$ revealing the roots evaluated at the monomial vectors $\boldsymbol{k}$ can be reconstructed from the basis of the null space we employed together with the eigenvectors of $\boldsymbol{A}$ as

$$\boldsymbol{K} \approx \boldsymbol{H}\boldsymbol{V}$$

and rescaling the columns such that the first row equals ones, leading to

$$\boldsymbol{K} = \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 4 & 9 \\ 8 & 27 \end{pmatrix},$$

from which the two solutions can be read off from the second row, and are again correctly retrieved as $x = 2$ and $x = 3$.

It must be noted that any basis for the null space of $\boldsymbol{M}$ can be used to phrase the eigenvalue problem and retrieve the roots. We have that $(S_1\boldsymbol{Z})^+ S_x\boldsymbol{Z} \neq (S_1\boldsymbol{H})^+ S_x\boldsymbol{H}$, but it can easily be verified that the eigenvalues of $(S_1\boldsymbol{Z})^+ S_x\boldsymbol{Z}$ and $(S_1\boldsymbol{H})^+ S_x\boldsymbol{H}$ will coincide, as they represent the common roots of the equations f_i.

4.2.3 Roots at Infinity and Descriptor Systems

When one considers univariate polynomials, roots at infinity can only occur when the coefficient of the highest-degree term is zero, and are for some reason considered explicitly. In normal circumstances, one would not easily consider a polynomial $p(x) = 5x^2 - 2x + 8$ as $p(x) = 0x^3 + 5x^2 - 2x + 8$. In the univariate case, this (somewhat artificial) case of the existence of roots at infinity will require notions of realization theory for descriptor systems, which turn out to have a natural interpretation in terms of the roots of univariate polynomials.

First of all, we will modify the definition of the annihilator of the Sylvester matrix, denoted by $\boldsymbol{V}(d)$.

Proposition 4.19 (Annihilator of Sylvester Matrix)**.** The Sylvester matrix $\boldsymbol{M}(d)$ is annihilated by a Vandermonde structured basis defined as

$$\boldsymbol{V}(d) := \left(\begin{array}{c|c} - \boldsymbol{v}_0^T\boldsymbol{A}^0 - & - \boldsymbol{w}_d^T\boldsymbol{E}^d - \\ - \boldsymbol{v}_0^T\boldsymbol{A}^1 - & \vdots \\ \vdots & - \boldsymbol{w}_d^T\boldsymbol{E}^1 - \\ - \boldsymbol{v}_0^T\boldsymbol{A}^d - & - \boldsymbol{w}_d^T\boldsymbol{E}^0 - \end{array}\right).$$

The left part represents the affine roots with the corresponding action matrix $\boldsymbol{A}$, and the right part represents the roots at infinity by means of the action matrix $\boldsymbol{E}$.

Recall from Chapter 3 that the matrix $\boldsymbol{E}$ is nilpotent and hence, for some μ we have that $\boldsymbol{E}^k = 0$, for $k \geq \mu$. In the case that there are roots at infinity, the considerations regarding the shift-invariance of above still hold. It is possible to retrieve the action matrices $\boldsymbol{A}$ and $\boldsymbol{E}$ from the (numerical basis for the) null space of the Sylvester matrix. However, one needs a mechanism for separating the affine roots and the roots at infinity.

In Figure 4.1 an overview of the proposed method is provided.

Sylvester Matrix **Null Space** **Dynamical System**

$$\left(\begin{array}{c} \boldsymbol{x}^{\sigma_1} \cdot f_1 \\ \hline \boldsymbol{x}^{\sigma_2} \cdot f_2 \\ \hline \vdots \\ \hline \boldsymbol{x}^{\sigma_n} \cdot f_s \end{array}\right) \Leftrightarrow \left(\begin{array}{c|c} \boldsymbol{v}_0^T \boldsymbol{A}^0 & \boldsymbol{w}_d^T \boldsymbol{E}^d \\ \hline \vdots & \vdots \\ \hline \boldsymbol{v}_0^T \boldsymbol{A}^d & \boldsymbol{w}_d^T \boldsymbol{E}^0 \end{array}\right) \Leftrightarrow \left\{\begin{array}{rcl} \boldsymbol{v}(k+1) & = & \boldsymbol{A}\boldsymbol{v}(k) \\ \boldsymbol{w}(k-1) & = & \boldsymbol{E}\boldsymbol{w}(k) \\ \boldsymbol{v}(0) & = & \boldsymbol{v}_0 \\ \boldsymbol{w}(d) & = & \boldsymbol{w}_d \end{array}\right.$$

Figure 4.1: Overview of realization theory for univariate polynomial root-finding. In the first step the Sylvester matrix $\boldsymbol{M}$ is built which contains the coefficients of the input equations f_i, for $i = 1, \ldots, s$. In the (right) null space of $\boldsymbol{M}$ the matrix $\boldsymbol{V}$ exhibits an observability matrix-like structure. From the structure of $\boldsymbol{V}$ the action matrices $\boldsymbol{A}$ and $\boldsymbol{E}$ can be obtained by solving systems of linear equations. Finally the linear dynamical system representation is found. In the case there are roots at infinity, this is a descriptor system.

Example 4.20. Let us revisit the system from Example 4.18 where we modify the equations such that there are roots at infinity. This is achieved by introducing leading zeros. Let us consider the equations

$$\begin{array}{rcl} 0x^4 + 0x^3 + x^2 - 5x + 6 & = & 0, \\ 0x^5 + 0x^4 + x^3 - 6x^2 + 11x - 6 & = & 0. \end{array}$$

The Sylvester matrix for degree $d = 5$ has as its canonical null space $\boldsymbol{H}$ which immediately allows separating the regular and singular parts as follows,

$$\boldsymbol{H} = \left(\begin{array}{c|c} \boldsymbol{H}_R & \boldsymbol{H}_S \end{array}\right) = \left(\begin{array}{rr|rr} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -6 & 5 & 0 & 0 \\ -30 & 19 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right).$$

Indeed, the leading zeros in the equations give rise to zero columns in $\boldsymbol{M}$ which give rise to unit columns in $\boldsymbol{H}$. From the regular part $\boldsymbol{H}_R$ the common roots $x = 2$ and $x = 3$ can be obtained as illustrated in Example 4.18. From the singular part $\boldsymbol{H}_S$ we can determine the action matrix $\boldsymbol{E}$ from the relation

$$\begin{pmatrix} \boldsymbol{E}^4 \\ \boldsymbol{E}^3 \\ \boldsymbol{E}^2 \\ \boldsymbol{E}^1 \\ \boldsymbol{E}^0 \end{pmatrix} \boldsymbol{E} = \begin{pmatrix} \boldsymbol{E}^5 \\ \boldsymbol{E}^4 \\ \boldsymbol{E}^3 \\ \boldsymbol{E}^2 \\ \boldsymbol{E}^1 \end{pmatrix},$$

or

$$\boldsymbol{E} = \overline{\boldsymbol{H}_S}^+ \underline{\boldsymbol{H}_S},$$

or

$$\boldsymbol{E} = (\boldsymbol{S}_x \boldsymbol{H}_S)^+ \boldsymbol{S}_1 \boldsymbol{H}_S,$$

leading to

$$\boldsymbol{E} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Notice that $\boldsymbol{E}^d = 0$ for $d \geq 2$.

The initial state for the regular part of the descriptor system is found as in Example 4.18 as $\boldsymbol{v}_0^T = \begin{pmatrix} 1 & 0 \end{pmatrix}$. For the singular part, the initial state (note that the singular part defines a backward running iteration) is found as

$$\boldsymbol{w}(d) = \begin{pmatrix} 0 & 1 \end{pmatrix}^T.$$

We can finally interpret the univariate root-finding problem as the descriptor form state space model (with $d := 5$)

$$\begin{aligned} \boldsymbol{v}(k+1) &= \begin{pmatrix} 0 & 1 \\ -6 & 5 \end{pmatrix} \boldsymbol{v}(k), & \boldsymbol{v}(0) &= \begin{pmatrix} 1 & 0 \end{pmatrix}^T \\ \boldsymbol{w}(k-1) &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \boldsymbol{w}(k), & \boldsymbol{w}(d) &= \begin{pmatrix} 0 & 1 \end{pmatrix}^T. \end{aligned}$$

Macaulay Matrix Formulation 5

The current chapter introduces the Macaulay matrix, which is based upon the work of Francis Sowerby Macaulay (Macaulay, 1902, 1916). This work was a generalization of Sylvester's resultant method to the multivariate case and it considered sub-resultants of the so-called Macaulay coefficient matrix. Although the core of this approach dates back to the end of the 19th and the beginning of the 20th century, due to historical reasons, these matrix-based methods have been largely neglected until the end of the 20th century.

This chapter starts off with defining the Macaulay matrix. Next several properties are studied, most of which have importance for the polynomial system solving problem. In particular, the (right) null space of the Macaulay matrix will turn out to be of paramount importance. We will discuss how the Macaulay matrix and its null space allow for an interpretation of monomials as being either linearly independent or linearly dependent on other monomials. This concept will play an important role in the linear algebra interpretation of the polynomial system solving problem. Furthermore, the shift-invariance property of the null space, which is prescribed by the monomial structure by which the Macaulay matrix is defined, will be discussed.

5.1 Representation of System of Polynomials

5.1.1 Problem Statement

We consider the problem of finding the solutions of a system of s multivariate polynomial polynomials f_i where $i = 1, \ldots, n$ in $n \leq s$ unknowns $x_1, \ldots, x_n$, having total degrees $d_1, \ldots, d_s$. The system of equations is represented formally as

$$\begin{cases} f_1(x_1,\ldots,x_n) & = & 0, \\ f_2(x_1,\ldots,x_n) & = & 0, \\ & \vdots & \\ f_s(x_1,\ldots,x_n) & = & 0. \end{cases} \tag{5.1}$$

The maximal total degree is denoted as $d_\circ = \max(d_1,\ldots,d_s)$. It is assumed that (5.1) describes a zero-dimensional solution set in the projective space with simple affine solutions. In most cases, we will consider the case $s = n$, however the theory is valid for the case $s > n$, provided that the system (5.1) has solutions. The case for which approximate solutions are useful to consider is discussed in Chapter 8.

5.1.2 Definition Macaulay Matrix

It is well-known that the space of multivariate polynomials up to a given degree d in n variables has the structure of a vector space. From the linear algebra perspective it is indeed natural to think of a polynomial as a vector containing its coefficients multiplied with a vector containing all possible monomials.

By multiplying all the equations f_i in (5.1) by monomials, polynomial equations are found which compose the rows of a so-called Macaulay matrix. This gives rise to a matrix equation of the form

$$\boldsymbol{M}(d)\,\boldsymbol{k}(d) = 0.$$

It will often turn out to be necessary to carefully order monomials, for which we have chosen to use the degree negative lexicographic ordering.[1]

Definition 5.1 (Degree Negative Lexicographic Order)**.** Let $\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{N}^n$ be monomial exponent vectors. Then two monomials represented by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are ordered by the degree negative lexicographic order as $\boldsymbol{\alpha} <_{\text{dnlex}} \boldsymbol{\beta}$ (simplified as $\boldsymbol{\alpha} < \boldsymbol{\beta}$), if

- $|\boldsymbol{\alpha}| < |\boldsymbol{\beta}|$, or
- $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}|$ and in the vector difference $\boldsymbol{\beta} - \boldsymbol{\alpha} \in \mathbb{Z}^n$, the left-most non-zero entry is negative.

Example 5.2. The monomials of maximal degree three in two variables x_1 and x_2 are ordered by the degree negative lexicographic order as

$$1 < x_1 < x_2 < x_1^2 < x_1x_2 < x_2^2 < x_1^3 < x_1^2x_2 < x_1x_2^2 < x_2^3.$$

[1] Most of the results in this thesis immediately hold for any graded monomial ordering.

Definition 5.3 (Macaulay matrix and monomial vector)**.** The Macaulay matrix $\boldsymbol{M}(d)$ contains as its rows the vector representations of $\boldsymbol{x}^{\sigma_i} \cdot f_i$, for all i, where $\boldsymbol{x}^{\sigma_i}$ represents a monomials having $\deg(\boldsymbol{x}^{\sigma_i}) \leq d$, and the columns of $\boldsymbol{M}(d)$ are indexed by all monomials of degree at most d, represented as follows,

$$\boldsymbol{M}(d) := \left(\begin{array}{c} \{\boldsymbol{x}^{\sigma_1}\} \cdot f_1 \\ \hline \{\boldsymbol{x}^{\sigma_2}\} \cdot f_2 \\ \hline \vdots \\ \hline \{\boldsymbol{x}^{\sigma_n}\} \cdot f_n \end{array} \right),$$

where each equation f_i, for $i = 1, \ldots, n$ is multiplied by all monomials of degrees $\leq d - d_i$, denoted by $\{\boldsymbol{x}^{\sigma_i}\}$. The rows of $\boldsymbol{M}$ are ordered by considering the monomials shifting f_i by the degree negative lexicographic order. The multivariate Vandermonde monomial vector $\boldsymbol{k}(d)$ is composed accordingly, *i.e.*,

$$\boldsymbol{k}(d) := \begin{pmatrix} 1 & x_1 & \ldots & x_n & x_1^2 & \ldots & x_n^2 & \ldots & x_1^d & \ldots & x_n^d \end{pmatrix}^T.$$

The Macaulay matrix construction leads to a very sparse and structured matrix where each row in the f_i-block contains only as many non-zero elements as there are non-zero coefficients in f_i.

Example 5.4. Consider the simple system

$$\begin{array}{rcl} f_1(x_1, x_2) & = & x_1^2 + x_1 x_2 + 4 \\ f_2(x_1, x_2) & = & 2x_1^3 + 2x_1 x_2^2 + 8, \end{array}$$

with $d_1 = 2$ and $d_2 = 3$. The Macaulay matrix $\boldsymbol{M}(4)$ is

$$\begin{array}{c|ccccccccccccccc} & 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 & x_2^4 \\ \hline f_1 & 4 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1f_1 & 0 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_2f_1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1^2f_1 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ x_1x_2f_1 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ x_2^2f_1 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ f_2 & 8 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1f_2 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ x_2f_2 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 \end{array},$$

where the first six rows correspond to the shifts of f_1 with all monomials of degrees up to $4 - d_1 = 2$ and the last three rows are the shifts of f_2 with all monomials of degrees up to $4 - d_2 = 1$.

5.1.3 Homogeneous Macaulay Matrix

In many cases, so-called roots at infinity occur. They are either caused by the presence of zero coefficients, or by the existence of algebraic relations among the coefficients in the equations. In order to describe the solutions at infinity, we use the well-known concepts of homogenization and projective space. The construction of the Macaulay matrix of the system (5.1) as in Definition 5.3 can (and *should*) always be interpreted as the Macaulay matrix of the homogenized system.

Let us therefore first review the definitions of homogenization and dehomogenization.

Definition 5.5 (Homogenization and dehomogenization)**.** The homogenization of an equation f, denoted f^h, is computed using the formula

$$f^h = x_0^d \cdot f\left(x_1/x_0, \ldots, x_n/x_0\right).$$

Dehomogenizing f^h yields f, or formally

$$f^h(1, x_1, \ldots, x_n) = f(x_1, \ldots, x_n).$$

A homogenized system of equations describes solutions in the $n+1$-dimensional projective space, and $x_0, \ldots, x_n$ are called the homogeneous coordinates. In the projective space the roots at infinity are incorporated as regular points for which $x_0 = 0$.

Definition 5.6 (Homogeneous Macaulay matrix and monomial vector)**.** The Macaulay matrix for the homogenized system $f_i^h(x_0, x_1, \ldots, x_n) = 0$, for $i = 1, \ldots, n$ is denoted by $\boldsymbol{M}^h(d)$ and is defined as

$$\boldsymbol{M}^h(d) := \left(\begin{array}{c} \{\boldsymbol{x}^{\sigma_1}\} \cdot f_1^h \\ \hline \{\boldsymbol{x}^{\sigma_2}\} \cdot f_2^h \\ \hline \vdots \\ \hline \{\boldsymbol{x}^{\sigma_n}\} \cdot f_n^h \end{array}\right),$$

where each equation f_i^h, for $i = 1, \ldots, n$ is multiplied by all monomials in the unknowns $x_0, \ldots, x_n$ of degree $d - d_i$, denoted by $\{\boldsymbol{x}^{\sigma_i}\}$. The columns of $\boldsymbol{M}^h(d)$ are indexed by all monomials in the unknowns $x_0, \ldots, x_n$ of degree d, which are placed in the multivariate Vandermonde monomial vector $\boldsymbol{k}^h(d)$ to obtain the equation

$$\boldsymbol{M}^h(d)\boldsymbol{k}^h(d) = 0.$$

The vector $\boldsymbol{k}^h(d)$ consists of all monomials in $n+1$ unknowns of degree d,

$$\boldsymbol{k}^h(d) := \begin{pmatrix} x_0^d & x_0^{d-1}x_1 & \ldots & x_0^{d-1}x_n & \ldots & x_1^d & \ldots & x_n^d \end{pmatrix}^T.$$

The following proposition will be very important when we are interpreting the results of our methods.

Proposition 5.7. The Macaulay matrix for the system of homogeneous equations $f_i^h(x_0, x_1, \ldots, x_n) = 0$, for $i = 1, \ldots, n$, is identical to the Macaulay matrix for the system (5.1), *i.e.*,

$$\boldsymbol{M}^h(d) \equiv \boldsymbol{M}(d).$$

It is easily seen that the equivalence of $\boldsymbol{M}(d)$ and $\boldsymbol{M}^h(d)$ lies in the mere relabeling of rows and columns of $\boldsymbol{M}(d)$. The ordering of the monomials in this relabeling is consistent with the degree negative lexicographic monomial ordering.

5.2 Properties

5.2.1 Number of Rows and Columns

The following formulas express the number of monomials (either of total degree d or of total degree $\leq d$) by binomial coefficient expressions. These expressions easily follow from Lemma B.3.

Lemma 5.8 (Dimensions Macaulay matrix)**.** Let $p(d)$ and $q(d)$ denote the number of rows and columns of $\boldsymbol{M}(d)$, respectively. We have

$$p(d) = \sum_{i=1}^{n} \binom{n + d - d_i}{d - d_i},$$

and

$$q(d) = \binom{n + d}{d}.$$

5.2.2 Structure and Sparsity

Density of Macaulay Matrix

The Macaulay matrix as defined in Definition 5.3 gives rise to a very sparse and structured matrix. The sparsity arises because in each row only as many nonzero elements occur as there are coefficients in the corresponding constituting polynomial f_i. The density of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of d is shown for a few combinations of n and $d_\circ$ in Figure 5.1. For example, the Macaulay matrix of a system of 5 equations with $d_\circ = 3$, the density drops below 1% at $d = 12$. Due to the Vandermonde structured basis vectors, and the fact that all rows are shifts of the equations f_i, the same coefficients occur in a very structured fashion.

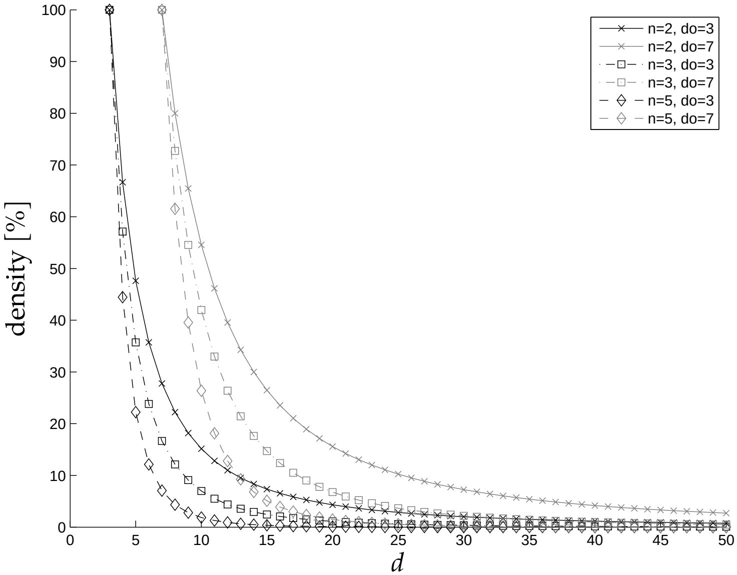

Figure 5.1: Density plot of Macaulay matrix $\boldsymbol{M}(d)$ for a few combinations of n and $d_\circ$ as a function of d. We see that the Macaulay matrix very quickly becomes very sparse. For example, the Macaulay matrix of a system of 5 equations with $d_\circ = 3$, the density drops below 1% at $d = 12$.

Alternative Definition

The Macaulay matrix can alternatively be constructed iteratively by considering increasing degrees. In every degree iteration, a number of new rows is added, corresponding to new shifts of equations that can be adjoined because a larger total degree is considered. It can easily be seen that the Macaulay matrix obtained in this way, denoted $\boldsymbol{N}(d)$ is related to the Macaulay matrix from Definition 5.3, denoted $\boldsymbol{M}(d)$ by a row permutation, *i.e.*,

$$\boldsymbol{N}(d) = \boldsymbol{P}\boldsymbol{M}(d).$$

Remark that the Macaulay matrix $\boldsymbol{N}(d)$ is defined such that it exhibits a 'nested' structure: This is a banded block structure with quasi-Toeplitz structure over the blocks. In $\boldsymbol{N}(d)$ the matrix $\boldsymbol{N}(d-1)$ occurs as a sub-matrix in the top-left part; in $\boldsymbol{N}(d-1)$ the matrix $\boldsymbol{N}(d-2)$ occurs as a sub-matrix, *etc.* Moreover, due to this structure and its construction, one can identify for every degree a number of nonzero blocks in which the coefficients of the polynomials f_i occur. These blocks are repeated in a quasi-Toeplitz structure: the blocks are repeated along the diagonals of $\boldsymbol{N}$. It is called a quasi-Toeplitz structure because the elements in the repeated blocks do not satisfy a strict Toeplitz structure, because for increasing degrees the blocks are growing in row and column dimensions.

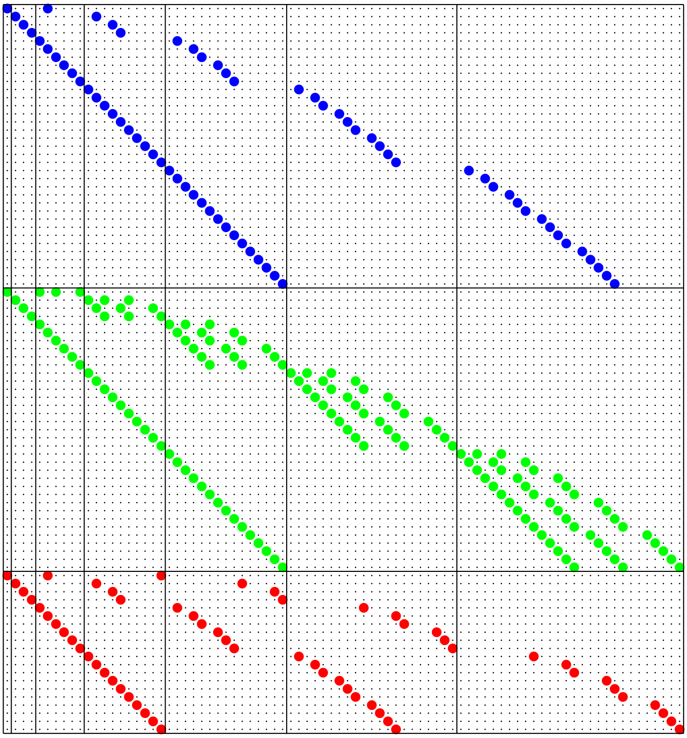

Figure 5.2: Sparsity plot of 'generalized Sylvester structured' Macaulay matrix $\boldsymbol{M}(6)$ of (5.2). We distinguish 3 blocks, each of which corresponds to one of the equations f_i and its shifts up to degree $d = 6$. The nonzero elements are represented by the blue (●), green (●) and red (●) bullets, corresponding to coefficients of the equations f_1, f_2 and f_3, respectively. The black dots $(\cdot)$ represent zero elements. The horizontal black lines mark the separation between three equation blocks. The vertical black lines denote the separation between the degree-blocks of the Macaulay matrix.

Example 5.9. Consider the system

$$\begin{array}{rcl} f_1(x_1,x_2,x_3) & = & x_1x_2 - 3 = 0, \\ f_2(x_1,x_2,x_3) & = & x_1^2 - x_3^2 + x_1x_3 - 5 = 0, \\ f_3(x_1,x_2,x_3) & = & x_3^3 - 2x_1x_2 + 7 = 0, \end{array} \tag{5.2}$$

with $d_1 = 2$, $d_2 = 2$ and $d_3 = 3$. The Macaulay matrix $\boldsymbol{M}(d)$ for degree $d = 6$, where the rows corresponding to an equation f_i occur in blocks, is represented in Figure 5.2. The quasi-Toeplitz structured Macaulay matrix $\boldsymbol{N}(d)$ for degree $d = 6$, in which the subsequent blocks are nested, is shown in Figure 5.3.

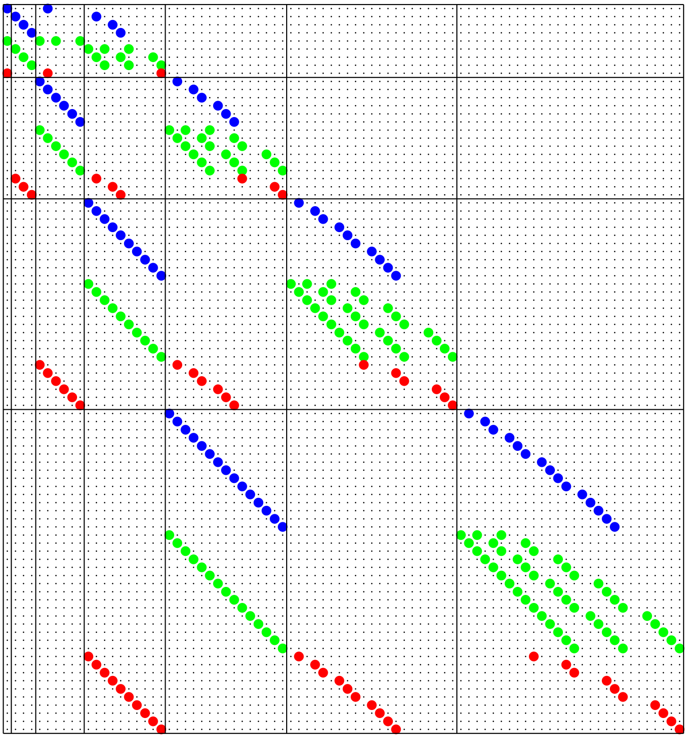

Figure 5.3: Sparsity plot of 'quasi-Toeplitz structured' Macaulay matrix $N(6)$ of (5.2). As the degree d increases, all 'new' shifts of the equations are adjoined to the previous iteration of the matrix. The nonzero elements are represented by the blue (•), green (•) and red (•) bullets, corresponding to coefficients of the equations f_1, f_2 and f_3, respectively. The black dots ($\cdot$) represent zero elements. The horizontal black lines mark the separation between the subsequent iterations d. The vertical black lines denote the separation between the degree-blocks of the Macaulay matrix.

5.3 Null Space of the Macaulay Matrix

The Macaulay matrix is a useful tool in computational algebraic geometry (Macaulay, 1902, 1916; Jónsson and Vavasis, 2004; Bondyfalat et al., 2000; Batselier et al., 2013b,a). For the problem of polynomial system solving, its null space is of particular interest. We will therefore employ the polynomial system solving problem as a starting point to describe its null space.

A central notion in the exposition is the distinction between linearly independent and linearly dependent monomials, which is closely related to the so-called set of standard monomials (also known as the normal set of the quotient space $\mathbb{C}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_s \rangle$ (see Appendix B). In the null space of the Macaulay matrix the linearly independent monomials correspond to the linearly independent rows. Given the complementarity property (Chapter 3), it is also expressed in the linearly dependent columns of the Macaulay matrix.

It will turn out that some of the standard monomials 'stabilize' as they are monitored for increasing degrees d, whereas others 'move along' to higher degrees. This will be linked to the so-called affine solutions and the so-called solutions at infinity. By monitoring the behavior of the standard monomials as the degree of the Macaulay matrix increases, the separation between the affine solutions and the solutions at infinity can be established.

We want to find as linearly independent monomials the monomials of lowest degrees possible. The complementarity/duality property of Chapter 3 tells us there are two ways to do this:

- We monitor the linearly *independent* rows as we adjoin rows in the null space of $\boldsymbol{M}$, going 'from the top to the bottom'. The linearly independent monomials correspond to the linearly independent rows.
- We monitor the linearly *dependent* columns in $\boldsymbol{M}$ going 'from the right to the left'. The linearly independent monomials are then the monomials corresponding to the linearly dependent columns of $\boldsymbol{M}$.

5.3.1 Generic Case: Affine Roots Only

We start with describing the case in which the input system (5.1) has only simple affine roots, which we call the generic case. This situation occurs *e.g.,* if all the possible coefficients in the system (5.1) occur as random numbers. Although the genericity assumption often does not always hold in practice, this case will be instrumental as a baseline setting for introducing the main ideas of the thesis.

Set of Standard Monomials

A central object in our exposition is the set of standard monomials. In the classical literature, the set of standard monomials (sometimes called the *normal set*) is defined as a basis of polynomial ring modulo the ideal generated by the system of polynomial equations, and is usually determined by means of Gröbner basis computations (Cox et al., 2005; Stetter, 2004), see Appendix B.

Here we define the standard monomials by means of numerical rank properties of the Macaulay matrix.

Definition 5.10 (Set of standard monomials and its complement)**.** The set of standard monomials $B(d)$ for degree d of the system (5.1) is defined as the complement of the monomials of degrees $\delta = 0, \ldots, d$ indexing the right-most linear independent columns of $\boldsymbol{M}(d)$. The complementary set $\overline{B(d)}$ is defined as the set of all monomials indexing the columns of $\boldsymbol{M}(d)$ that are not standard monomials.

A rudimentary numerical procedure for numerically determining the standard monomials is to iterate over the columns of $\boldsymbol{M}(d)$ and monitor increases in the (numerical) rank as more columns are added, starting from the right-most column. The columns which leave the rank unchanged are in the set of standard monomials $B(d)$, whereas columns which increase the rank are in the complement of the set of standard monomials $\overline{B(d)}$.[2]

Remember from the duality property between the indices of the columns of a matrix and the rows of its null space of Section 3.1. This allows us to alternatively interpret the standard monomials as the 'first' linearly independent rows of a basis for the null space of the Macaulay matrix. The notion of standard monomials is illustrated using an example.

Example 5.11. We continue with the equations from Example 5.4. Let the standard monomials $B(d)$ be monomials indexing the right-most linearly dependent columns as in the definition. One can easily check that the right-most linearly dependent columns of $\boldsymbol{M}(4)$ are the columns corresponding to the monomials $x_2^4, x_2^3, x_1^2, x_2, x_1$ and 1.

Alternatively, by computing a basis for the null space of $\boldsymbol{M}(d)$, *e.g.*, using SVD, and inspecting which rows (starting from the top) contribute to the rank, we also find $1, x_1, x_2, x_1^2, x_2^3, x_2^4$.

Hence, we have

$$B(4) = \{1, x_1, x_2, x_1^2, x_2^3, x_2^4\}.$$

Multivariate Vandermonde Null Space of the Macaulay Matrix

From the Macaulay matrix construction it immediately follows that each solution of (5.1), denoted as $\boldsymbol{x}^{(i)} := \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\right)$, for $i = 1, \ldots, m_B$, composes a vector in the null space. The multivariate Vandermonde null space is defined as the collection of all such vectors.

Definition 5.12 (Multivariate Vandermonde null space)**.** Evaluating $\boldsymbol{k}$ at the m_B solutions gives rise to m_B independent vectors in the null space of $\boldsymbol{M}$. We

[2] Numerical rank tests are implemented in a numerically reliable way by the SVD. Either a suitable threshold value is required for deciding whether a singular value is small enough to be considered as zero, or a sufficiently large decay in consecutive singular values needs to be observed.

denote by $\boldsymbol{k}|_{\boldsymbol{x}^{(i)}}$, for $i = 1, \ldots, m_B$ the evaluation of $\boldsymbol{k}$ at one of the m_B solutions. The multivariate Vandermonde null space of $\boldsymbol{M}$ is defined as the collection of the vectors $\boldsymbol{k}|_{\boldsymbol{x}^{(i)}}$into the multivariate Vandermonde structured matrix $\boldsymbol{K}$ of size $q \times m_B$, *i.e.*,

$$\boldsymbol{K} := \begin{pmatrix} | & | & | \\ \cdots & \boldsymbol{k}|_{\boldsymbol{x}^{(i)}} & \cdots \\ | & | & | \end{pmatrix}.$$

The multivariate Vandermonde null space $\boldsymbol{K}$ will only be used in the derivations and will not be constructed explicitly (it can only be constructed when the solutions are known priorly!). Therefore, the specific order in which the vectors $\boldsymbol{k}$ are placed in $\boldsymbol{K}$ is not of any relevance for the remainder of our exposition.

Example 5.13. Let us consider a small system of polynomial equations

$$\begin{array}{rcccl} f_1(x_1, x_2) & = & x_2 - x_1^2 & = & 0, \\ f_2(x_1, x_2) & = & x_2 - 2x_1 & = & 0, \end{array}$$

which has the solutions $(0,0)$ and $(2,4)$. The Macaulay matrix for degree $d = 2$ is

$$\boldsymbol{M}(2) = \left(\begin{array}{c|cc|ccc} 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 1 \end{array}\right).$$

The multivariate Vandermonde null space is

$$\boldsymbol{K}(2) = \left(\begin{array}{c|c} 1 & 1 \\ \hline 0 & 2 \\ 0 & 4 \\ \hline 0 & 4 \\ 0 & 8 \\ 0 & 16 \end{array}\right).$$

The solutions are in the second and third row of $\boldsymbol{K}(2)$.

The following lemmas describe well-known facts regarding the number of standard monomials, the number of solutions of the system (5.1). In the next section, we will relate this to the nullity of the Macaulay matrix.

Lemma 5.14 (Bézout number m_B (Cox et al., 2005))**.** The number of solutions of (5.1) in projective space (counting multiplicities) is given by the Bézout number

$$m_B = \prod_{i=1}^{n} d_i.$$

We will relate the Bézout number to the nullity of the Macaulay matrix, but we first need to discuss the issue of roots at infinity.

5.3.2 Solutions at Infinity

In many cases, the genericity assumption of Section 5.3.1 that the solution set contains only affine roots does not hold. In addition to affine solutions, so-called solutions at infinity may exist, which are caused by algebraic relations between the coefficients or the occurrence of zero coefficients in (5.1). Not surprisingly, these algebraic relations will be manifested by rank-deficiencies in the Macaulay matrix.

The following example illustrates how solutions at infinity can show up in a very simple system of polynomial equations.

Example 5.15. Consider the equations

$$\begin{array}{rcccl} f_1(x_1, x_2) & = & x_1^2 - x_2 & = & 0, \\ f_2(x_1, x_2) & = & x_1 - 5 & = & 0, \end{array}$$

geometrically represented by the intersection of a vertical line and a parabola (Figure 5.4). We can see that there is only a single solution, which can easily be confirmed by substituting the value $x_1 = 5$ into the first equation, leading to the solution $(5, 25)$. The Bézout number of this system predicts that there are two solutions, and, as a matter of fact, it turns out that there are two solutions, the first one is the affine solution $(5, 25)$, and the second one is a so-called solution at infinity.

The solution at infinity can be described using the homogenized system of equations

$$\begin{array}{rcccl} f_1^h(x_0, x_1, x_2) & = & x_1^2 - x_0 x_2 & = & 0, \\ f_2^h(x_0, x_1, x_2) & = & x_1 - 5x_0 & = & 0. \end{array}$$

We can now write the solutions of the homogenized system as the triplets (x_0, x_1, x_2) (hereby keeping in mind that points in the projective space are scaling invariant). We find two projective solutions as $(0, 0, 1)$ and $(1, 5, 25)$. We recognize the affine solution $(5, 25)$, but additionally, there is a solution for which $x_0 = 0$. This is the solution at infinity. Notice that 'dehomogenizing' would lead to a singularity for the solution at infinity: one would need to compute $x_1/0$ and $x_2/0$.

Lemma 5.16 (Nullity of Macaulay matrix equals Bézout number)**.** *At a sufficiently large degree d_c, the nullity of the Macaulay matrix is equal to the Bézout number, i.e.,*

$$\text{nullity}\,(\boldsymbol{M}(d)) = \prod_{i=1}^{n} d_i, \quad d \geq d_c.$$

Proof. The cardinality of the set of standard monomials $B(d_c)$ as defined in Definition 5.10 is equal to the nullity of $\boldsymbol{M}$, which immediately follows from the definition. From Cox et al. (2005) it follows that this number corresponds

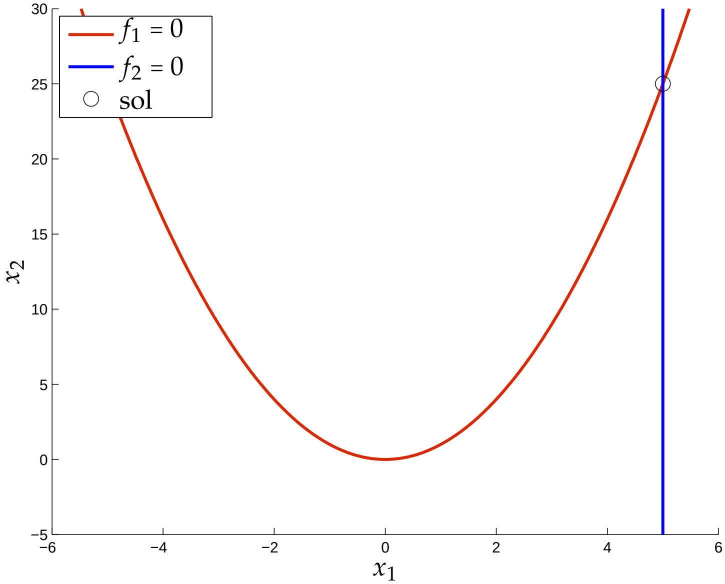

Figure 5.4: Geometrical representation of Example 5.15. The Bézout number of the defining system is $m_B = 2$, however, there is only a single affine root. The second root is a so-called root at infinity.

to the dimension of the quotient space $\mathbb{C}[x_1, \ldots, x_n]/\langle f_1, \ldots, f_n \rangle$ in the generic case. The same reasoning holds for the projective case. □

Example 5.17. Let us revisit the equations from Example 5.15. The Macaulay matrix for the system at degree $d = 2$ is constructed as

$$M(2) = \begin{array}{c|cc|ccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ -5 & 1 & 0 & 0 & 0 & 0 \\ 0 & -5 & 0 & 1 & 0 & 0 \\ 0 & 0 & -5 & 0 & 1 & 0 \end{array},$$

and we can assert that $\text{nullity}(\boldsymbol{M}(2)) = 2$.

An important observation can be made in this example. Recall that solutions at infinity are solutions for which $x_0 = 0$. This is expressed in the Macaulay matrix as a rank-deficiency in the block of the highest degree. Indeed, given that there are solutions for which $x_0 = 0$, the homogeneous interpretation of the Macaulay matrix reduces all columns of degrees lower than d to zero. Hence, in order for a solution to exist, there *must* be a rank-deficiency in the block built from the columns corresponding to the monomials of degree d. In this case, this can easily be checked visually: the column indexed by the monomial x_2^2 is zero.

The set of affine standard monomials

From the previous paragraphs we learn that solutions at infinity are defined as non-zero solutions for which the homogenization variable $x_0 = 0$. Lemma 5.16 and Proposition 5.7 imply that the solutions at infinity compose vectors in the null space of $\boldsymbol{M}(d_c)$.

Proposition 5.18. Consider the partitioning of $\boldsymbol{M}(d_c)$ as

$$\boldsymbol{M}(d_c) = \begin{pmatrix} \boldsymbol{M}_0 & \boldsymbol{M}_1 & \boldsymbol{M}_2 & \dots & \boldsymbol{M}_{d_c} \end{pmatrix},$$

where the block $\boldsymbol{M}_i$ contains the columns of $\boldsymbol{M}(d_c)$ indexed by the monomials of degree i, for $i = 0, \dots, d_c$. The existence of solutions at infinity is revealed by the column rank deficiency of the block $\boldsymbol{M}_{d_c}$.

Column rank deficiency of $\boldsymbol{M}_{d_c}$ implies there exist solutions for which the homogenization variable x_0 is zero. Also observe that evaluating $x_0 = 0$ would reduce all blocks $\boldsymbol{M}_d$ for $d < d_c$ to zero. An immediate consequence is that when determining the standard monomials $B(d)$ for a sufficiently large degree d, we will find some monomials that are caused by the non-zero solutions having $x_0 = 0$.[3]

Corollary 5.19. If $n = s$, the degree d_c is given by the expression

$$d_c = d^\star := \sum_{i=1}^{n} d_i - n + 1.$$

Proof. When there are no roots at infinity, all columns of $d^\star$ should be 'reachable' by shifts of the original equations. Without loss of generality, we can reason that in each of the equations there is a term x^{d_i} that serves as a pivot term that is able to reach one of the columns of $\boldsymbol{M}_{d^\star}$, it immediately follows that $d > d^\star$ is necessary to detect the rank-deficiency of $\boldsymbol{M}_{d^\star}$. Consequently, if not all columns of $\boldsymbol{M}_{d^\star}$ are linearly independent at $d^\star$, they will remain linearly dependent at degrees $d > d^\star$. □

The notion of the solutions at infinity implies the concept of the *affine standard monomials*. We will define this using the observation that as the degree d of the Macaulay matrix increases, the linear independent standard monomials of the affine solutions and the solutions at infinity become separated. At a certain degree, which we will call d_G, there is a sufficient separation between the two sets of monomials, and the monomials that stabilize are called the affine standard monomials.

Let us first reconsider the example of Section 2.3 to fix the ideas.

[3] Note, however, that it would in general not suffice to simply dismiss the standard monomials of degree d alone (and, *e.g.*, remove the corresponding columns from $\boldsymbol{M}(d)$) to resolve the solutions at infinity, since they may have — and often *do have* — an intricate multiplicity structure that protrudes into degrees smaller than d.

Example 5.20. Consider the simple system of two equations

$$\begin{array}{rcrcl} f_1(x_1,x_2) & = & x_1^2+x_1x_2-2 & = & 0, \\ f_2(x_1,x_2) & = & x_2^2+x_1x_2-2 & = & 0. \end{array}$$

We construct for several iterations the Macaulay matrix and monitor its rank, nullity and the indices of the linear independent monomials. The results are summarized in Table 5.1.

Table 5.1: Diagram showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d. The rank keeps increasing as d grows, however the nullity stabilizes at the value 4. Observe that only two of the linear independent monomials stabilize, namely 1 and x_1, whereas the remaining two shift towards higher degrees as the overall degree of the Macaulay matrix increases.

d	size $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	standard monomials **(affine)**
2	2×6	4	$1,x_1,x_2,x_1^2$
3	6×10	4	$1,x_1,x_1^2,x_1^3$
$4 =: d_G$	12×15	4	$\mathbf{1},\mathbf{x_1},x_1^3,x_1^4$
5	20×21	4	$\mathbf{1},\mathbf{x_1},x_1^4,x_1^5$
6	30×28	4	$\mathbf{1},\mathbf{x_1},x_1^5,x_1^6$

We observe that there are four linear independent monomials in all iterations, but only the monomials 1 and x_1 'stabilize', whereas the other two monomials are replaced by higher degree monomials as d increases. Observe that there is a pattern in the two remaining monomials: they are always given by x_1^d and x_1^{d-1}. It turns out that the system has two affine roots and two solutions at infinity. The affine roots will correspond to the monomials 1 and x_1 and the solutions at infinity correspond to the monomials at higher degrees.

By homogenizing the equations we can analyze the solutions at infinity. We find

$$\begin{array}{rcrcl} f_1^h(x_0,x_1,x_2) & = & x_1^2+x_1x_2-2x_0^2 & = & 0, \\ f_2^h(x_0,x_1,x_2) & = & x_2^2+x_1x_2-2x_0^2 & = & 0. \end{array}$$

We set $x_0 = 0$ and identify $x_1 + x_2$ as a common factor in both equations, confirming that there exists a solution at infinity, which is be described by $(x_0,x_1,x_2) = (0,\alpha,-\alpha)$.

Observe that the existence of solutions at infinity is also expressed in the Macaulay matrix. If there can be found linear independent monomials of degree d in $\boldsymbol{M}(d)$, for any sufficiently large degree d, there are solutions at infinity. Indeed, setting the homogenization variable x_0 to zero in the homogenized system is equivalent to retaining only the highest degree columns of the Macaulay matrix. Hence, if there is linear dependence among these columns, there are solutions at infinity.

The dynamical behavior of the structure of the null space as a function of d, when there are solutions at infinity, is revealed by inspection of the canonical basis for the null space, denoted by $\boldsymbol{H}(d)$. At degree $d = 4 =: d_G$ we clearly see the separation emerging between the affine roots and the solutions at infinity. At degree $d_G + 1 = 5$ the separation between the affine roots and the solutions at infinity is increased by one degree block, as shown in $\boldsymbol{H}(4)$ and $\boldsymbol{H}(5)$:

$$\boldsymbol{H}(4) \quad = \quad \begin{array}{c} \begin{array}{cc} \overbrace{\qquad}^{\text{affine}} & \overbrace{\qquad}^{\text{infinity}} \end{array} \\ \left(\begin{array}{cc|cc} \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & \mathbf{0} & \mathbf{0} \\ 1 & 0 & \mathbf{0} & \mathbf{0} \\ 1 & 0 & \mathbf{0} & \mathbf{0} \\ \hline 0 & 0 & \mathbf{1} & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & -1 & 0 \\ \hline 0 & 0 & 0 & \mathbf{1} \\ 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array} \begin{array}{c} \uparrow \\ \text{gap} \\ \downarrow \end{array} \quad \text{and} \quad \begin{array}{c} \begin{array}{cc} \overbrace{\qquad}^{\text{affine}} & \overbrace{\qquad}^{\text{infinity}} \end{array} \\ \left(\begin{array}{cc|cc} \mathbf{1} & 0 & 0 & 0 \\ \hline 0 & \mathbf{1} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & \mathbf{0} & \mathbf{0} \\ 1 & 0 & \mathbf{0} & \mathbf{0} \\ 1 & 0 & \mathbf{0} & \mathbf{0} \\ \hline 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ 0 & 1 & \mathbf{0} & \mathbf{0} \\ \hline 0 & 0 & \mathbf{1} & 0 \\ 2 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & \mathbf{1} \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & -1 \end{array}\right) \end{array} \begin{array}{c} \uparrow \\ \\ \text{gap} \\ \\ \downarrow \end{array} \quad = \quad \boldsymbol{H}(5).$$

In the canonical basis, as a function of the degree d, we see the appearance of zeros in the top part corresponding to the degrees 0, 1, 2, *etc.* of the columns 3 and 4. As d increases, a gap between the linear independent monomials corresponding to the affine roots and the linear independent monomials corresponding to the solutions at infinity emerges. We will therefore employ this phenomenon to separate the affine roots and the solutions at infinity.

A formal definition of the set of affine standard monomials is now provided.

Definition 5.21 (Set of affine standard monomials)**.** We call $B^{\star}(d_G) \subseteq B(d_G)$ the *affine standard monomials* for degree d_G. The degree d_G is the smallest degree for which for all $\boldsymbol{x}^{\boldsymbol{\alpha}} \in B^{\star}(d_G)$ and $\boldsymbol{x}^{\boldsymbol{\beta}} \in B(d_G) \backslash B^{\star}(d_G)$, we have $|\boldsymbol{\beta} - \boldsymbol{\alpha}| \geq 1$, where $\cdot\backslash\cdot$ denotes the set difference operator.

The definition leads to a procedure for determining the affine standard monomials: One inspects the elements in $B(d)$ as the degree d increases. The standard monomials corresponding to the affine roots will 'stabilize', whereas the ones corresponding to the solutions at infinity will always appear at high degrees (and move along as d increases). From a sufficiently large degree d_G on, a degree-gap arises between the set of affine standard monomials and the remaining standard monomials. A numerically reliable way to do this is to monitor the (numerical) rank increases by considering growing degree-blocks in a basis for the null space of $\boldsymbol{M}$ (or the columns of $\boldsymbol{M}$ going from right to left).

An interesting implication of the above is that the number of affine roots of (5.1) equals the number of affine standard monomials.

Lemma 5.22 (Number of affine roots (Batselier, 2013))**.** The number of affine standard monomials corresponds to the number of affine roots, or

$$m_a = \#B^\star(d_G),$$

where $\#(\cdot)$ denotes the set cardinality.

5.3.3 Removing the Solutions at Infinity

Assume that the Macaulay matrix for degree d_G is constructed and the set of affine standard monomials $B^\star(d_G)$ is determined. The definition of the affine standard monomials leads to two methods for discarding the solutions at infinity.

1. By removing from $\boldsymbol{M}(d_G)$ the standard monomial columns of the highest degrees, *i.e.*, the monomials $B(d_G)\backslash B^\star(d_G)$, the solutions at infinity are annihilated. This leads to a reduced Macaulay matrix $\boldsymbol{M}^\star(d_G)$.
2. One can also reason that the construction of the Macaulay matrix $\boldsymbol{M}(d)$ for increasing degrees d 'separates' the affine solutions and the solutions at infinity as suggested in Definition 5.21. This observation can be employed to search for the affine roots only when devising a root-finding algorithm (leading to the column compression alternative in Section 6.2.4).

5.3.4 Multiple Roots and the Dual Space

The Bézout number (Lemma 5.14) counts the number of solutions in the projective space, including multiplicities. However, the evaluation of the monomial vector $\boldsymbol{k}$ at a solution $\boldsymbol{x}^\star$ with multiplicity μ will only produce a single linearly independent vector in the null space of $\boldsymbol{M}$.

In the univariate case it is well-known that $f(x)$ has an μ-fold root $x^\star$ if and only if $(d^k f/dx^k)(x^\star) = 0$, for $k = 0, 1, \ldots, \mu - 1$ and $(d^\mu f/dx^\mu)(x^\star) \neq 0$.

The generalization of this notion to the multivariate case is attributed to Gröbner (Marinari et al., 1996). The interested reader is referred to Dayton et al. (2011); Marinari et al. (1996); Möller and Stetter (1995); Mourrain and Pan (2000) for a thorough study of the multiplicity structure in the multivariate case. We will illustrate the main ideas by means of an example, but will not go into details as the analysis of the multiplicity structure of the roots is not of direct relevance for the remainder of this manuscript.

The dual space theory allows to describe the relation between the multiplicity structure of the roots and the composition of the multivariate Vandermonde null space $\boldsymbol{K}(d)$ by means of linear combinations of partial derivatives of the monomial vectors $\boldsymbol{k}(d)$. Let

$$\partial_{\boldsymbol{\alpha}}\big|_{\boldsymbol{x}^\star} = \partial_{x_0^{\alpha_0} x_1^{\alpha_1} \ldots x_n^{\alpha_n}}\Big|_{\boldsymbol{x}^\star} := \frac{1}{\alpha_0!\,\alpha_1!\cdots\alpha_n!}\,\frac{\partial^{\alpha_0+\alpha_1+\ldots+\alpha_n}}{\partial x_0^{\alpha_0}\partial x_1^{\alpha_1}\ldots\partial x_n^{\alpha_n}}\Bigg|_{\boldsymbol{x}^\star}$$

be the partial differential operators acting on the monomial vectors $\boldsymbol{k}$.

Example 5.23. For the case $n = 2$ and $d = 3$ and derivatives up to degree 2 we have the following partial derivatives:

$$\left(\begin{array}{c|cc|ccc} | & | & | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & \partial_{20} & \partial_{11} & \partial_{02} \\ | & | & | & | & | & | \end{array}\right)$$

$$= \left(\begin{array}{r|rr|rrr} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline x_1 & 1 & 0 & 0 & 0 & 0 \\ x_2 & 0 & 1 & 0 & 0 & 0 \\ \hline x_1^2 & 2x_1 & 0 & 1 & 0 & 0 \\ x_1x_2 & x_2 & x_1 & 0 & 1 & 0 \\ x_2^2 & 0 & 2x_2 & 0 & 0 & 1 \\ \hline x_1^3 & 3x_1^2 & 0 & 3x_1 & 0 & 0 \\ x_1^2x_2 & 2x_1x_2 & x_1^2 & x_2 & 2x_1 & 0 \\ x_1x_2^2 & x_2^2 & 2x_1x_2 & 0 & 2x_2 & x_1 \\ x_2^3 & 0 & 3x_2^2 & 0 & 0 & 3x_2 \end{array}\right), \qquad (5.3)$$

where the horizontal bars indicate the degree blocks of the monomials and the vertical bars indicate the degrees of the derivatives. The columns are indexed by a simplified notation of the partial derivatives $\partial_{\boldsymbol{\alpha}}$.

The dual space $\boldsymbol{K}(d)$ is defined as a collection of linear combinations of such partial derivatives, evaluated at a solution of the systems, *i.e.,*

$$\boldsymbol{K}(d) = \left(\begin{array}{ccc} | & | & | \\ \cdots & \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}}\, \partial_{\boldsymbol{\alpha}}\big|_{\boldsymbol{x}^\star} & \cdots \\ | & | & | \end{array}\right),$$

where $c_{\boldsymbol{\alpha}} \in \mathbb{C}$ and $\partial_{\boldsymbol{\alpha}} f_i\big|_{\boldsymbol{x}^\star} = 0$, for all $i = 1, \ldots, n$.

Let us now consider an example where the multiplicity structure of a root is unraveled.

Example 5.24. Consider the equations

$$\begin{array}{rclcl} f_1(x_1,x_2) & = & (x_2-2)^2 & = & x_2^2-4x_2+4 & = & 0, \\ f_2(x_1,x_2) & = & (x_1-x_2+1)^2 & = & x_1^2-2x_1x_2+2x_1+x_2^2-2x_2+1 & = & 0, \end{array}$$

having a single solution $(1,2)$ with multiplicity 4 as we can easily understand from the equations.

We build the Macaulay matrix $\boldsymbol{M}(d)$ for degree $d=3$:

$$\boldsymbol{M}(3)=\left(\begin{array}{c|cc|ccc|cccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\ 4 & 0 & -4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & -2 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 4 & 0 & 0 & -4 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 & -2 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 & -2 & 0 & 1 & -2 & 1 \end{array}\right),$$

having a 4-dimensional null space, as prescribed by the Bézout number $m_B = 2\cdot 2$. The evaluation of $\boldsymbol{x}^\star=(1,2)$ at the multivariate Vandermonde vector $\boldsymbol{k}(3)$ will only constitute a single vector in the null space of the Macaulay matrix. It can be verified that the following vectors are linearly independent and are all null space vectors of $\boldsymbol{M}$:

$$\left(\begin{array}{cccc} | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ | & | & | & | \end{array}\right)$$

$$=\left(\begin{array}{cccc} \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ 1 & 0 & 0 & 0 \\ \hline x_1 & 1 & 0 & 0 \\ x_2 & 0 & 1 & 0 \\ \hline x_1^2 & 2x_1 & 0 & 2 \\ x_1x_2 & x_2 & x_1 & 1 \\ x_2^2 & 0 & 2x_2 & 0 \\ \hline x_1^3 & 3x_1^2 & 0 & 6x_1 \\ x_1^2x_2 & 2x_1x_2 & x_1^2 & 2x_2+2x_1 \\ x_1x_2^2 & x_2^2 & 2x_1x_2 & 2x_2 \\ x_2^3 & 0 & 3x_2^2 & 0 \end{array}\right).$$

Evaluated at the root $(1,2)$ we find

$$\boldsymbol{K}(3) = \begin{array}{c} \begin{array}{cccc} \partial_{00}|_{(1,2)} & \partial_{10}|_{(1,2)} & \partial_{01}|_{(1,2)} & (2\partial_{20}+\partial_{11})|_{(1,2)} \end{array} \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ \hline 1 & 2 & 0 & 2 \\ 2 & 2 & 1 & 1 \\ 4 & 0 & 4 & 0 \\ \hline 1 & 3 & 0 & 6 \\ 2 & 4 & 1 & 6 \\ 4 & 4 & 4 & 4 \\ 8 & 0 & 12 & 0 \end{array}\right) \end{array}.$$

5.3.5 Nullity of Macaulay Matrix and Dimension of Variety

An interesting link between the nullity of the Macaulay matrix and the dimension of the solution space of its constituting equations can be made. This exposition requires some notions from algebraic geometry. For background information the interested reader is referred to Appendix B.

For a sufficiently large degree d, we can write

$$\begin{aligned} \operatorname{nullity} \boldsymbol{M}(d) &= q(d) - \operatorname{rank} \boldsymbol{M}(d), \\ &= \dim \mathbb{C}[x_0, \ldots, x_n]_d - \dim\langle f_1^h, \ldots, f_s^h \rangle_d, \\ &= \dim \mathbb{C}[x_0, \ldots, x_n]_d / \langle f_1^h, \ldots, f_s^h \rangle_d, \\ &= \dim \mathbb{C}[x_0, \ldots, x_n]_d / I_d^h, \end{aligned}$$

where $I_d^h := \mathbb{C}[x_0, \ldots, x_n]_d \cap I^h$. Remark that this is closely related to the Hilbert polynomial (Cox et al., 2007):

$$\begin{aligned} {}^{a}HP_I(d) &= \dim \mathbb{C}[x_1, \ldots, x_n]_{\leq d} / I_{\leq d} \\ &= \dim \mathbb{C}[x_0, \ldots, x_n]_d / I_d^h \\ &= HP_{I^h}(d), \end{aligned}$$

where $I := \langle f_1, \ldots, f_s \rangle$ and $I^h := \langle f_1^h, \ldots, f_s^h \rangle$.

The Hilbert polynomial can be used to define the dimension of the corresponding variety: its degree corresponds to the dimension of the variety, *i.e.*, $\dim V = \deg HP$.

Hence, by monitoring the increases in the nullity of the Macaulay matrix, the dimension of the solution set can be determined: when the nullity of $\boldsymbol{M}(d)$ stabilizes to a constant, the solution set is zero-dimensional; when the nullity increases linearly with d, the solution set is one-dimensional, *etc.*

Remark 5.25. Keep in mind that the Macaulay matrix *always* describes the *projective* solution set: as we have discussed earlier, the Macaulay matrix is always (implicitly) operating in the projective coordinates $x_0, \ldots, x_n$.

Remark 5.26. It may happen that the affine variety is zero-dimensional, while the projective solution set is one-dimensional. As long as the affine standard monomials can be determined correctly this case will not pose any problems for the algorithms we will develop in the next chapter (see, *e.g.*, Example 6.14).

Polynomial System Solving Algorithms 6

In the previous chapter we have introduced the Macaulay matrix and studied its properties. It was shown that the null space is closely related with the solutions of the system (5.1). In this chapter we will use this insight and use a numerical basis of the null space as a tool for *computing* the roots.

First of all it is shown how the multiplication structure in the null space of the Macaulay matrix leads to an eigendecomposition. This immediately leads to the first algorithm operating in the null space of the Macaulay matrix. Also a procedure to break down the computation of the null space into iterative steps is developed.

Next the complementarity property between the rows of the null space and the columns of the Macaulay matrix is used to rewrite the system solving method such that the null space does not need to be computed. The second algorithm phrases the polynomial system solving problem as an eigenvalue problem by means of a column-repartitioned Macaulay matrix. The numerical implementation of this procedure is by means of a Q-less QR decomposition.

In both approaches the case of solutions at infinity is discussed and several numerical examples illustrating the operation of the algorithms are provided. As an application of these methods we will discuss a problem from system identification, namely the structured total least squares problem.

6.1 From Multiplication Structure to Eigenvalues

In a similar fashion as in the univariate case described in Section 4.1.2, multiplication of a multivariate Vandermonde monomial vector $\boldsymbol{k}$ by a monomial or a polynomial obeys a shift property which is an essential ingredient for phrasing the root-finding problem as an eigenvalue problem.

Proposition 6.1 (Monomial Shift in Monomial Vectors $\boldsymbol{k}$)**.** Multiplication of entries in a multivariate Vandermonde monomial vector $\boldsymbol{k} := \boldsymbol{k}(d)$ with the monomial $\boldsymbol{x}^\gamma$ maps the entries of $\boldsymbol{k}$ of degrees 0 up to $d - |\gamma|$ to entries in $\boldsymbol{k}$ of degrees $|\gamma|$ up to d. This is expressed by means of row selection matrices operating on $\boldsymbol{k}$ as

$$\boldsymbol{S}_1 \boldsymbol{k} \boldsymbol{x}^\gamma = \boldsymbol{S}_\gamma \boldsymbol{k},$$

where $\boldsymbol{S}_1$ selects all monomials in $\boldsymbol{k}$ of degrees 0 up to $d - |\gamma|$ and $\boldsymbol{S}_\gamma$ selects the rows of $\boldsymbol{k}$ onto which the monomials $\boldsymbol{S}_1\boldsymbol{k}$ are mapped by multiplication by $\boldsymbol{x}^\gamma$.

Example 6.2. Consider a multivariate Vandermonde monomial vector $\boldsymbol{k}$ of degree three in two variables x_1 and x_2, given by

$$\boldsymbol{k} = \left(1 \mid x_1 \quad x_2 \mid x_1^2 \quad x_1x_2 \quad x_2^2 \right)^T,$$

and a shift monomial x_1. We can write

$$\begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} x_1 = \begin{pmatrix} x_1 \\ x_1^2 \\ x_1x_2 \end{pmatrix},$$

which can alternatively be expressed as $\boldsymbol{S}_1 \boldsymbol{k} x_1 = \boldsymbol{S}_{x_1} \boldsymbol{k}$, with

$$\boldsymbol{S}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and}$$

$$\boldsymbol{S}_g = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

The same can be done for x_2.

This property can be generalized directly to an arbitrary polynomial shift function $g(\boldsymbol{x})$ with $\deg(g) \leq d$.

Proposition 6.3 (Polynomial Shift in Monomial Vectors $\boldsymbol{k}$)**.** Multiplication of a multivariate Vandermonde monomial vector $\boldsymbol{k}(d)$ with a polynomial $g(\boldsymbol{x}) := \sum_\gamma c_\gamma \boldsymbol{x}^\gamma$ gives $\boldsymbol{S}_1 \boldsymbol{k} g(\boldsymbol{x}) = \boldsymbol{S}_g \boldsymbol{k}$, where $\boldsymbol{S}_g := \sum_\gamma c_\gamma \boldsymbol{S}_\gamma$, in accordance with Proposition 6.1. The selection matrix $\boldsymbol{S}_g$ takes in this case linear combinations of rows of $\boldsymbol{k}$. A consequence is that any shift function g can be composed by shifting 'up' a single degree block.

Example 6.4. Consider a multivariate Vandermonde monomial vector $\boldsymbol{k}$ of degree three in two variables x_1 and x_2, given by

$$\boldsymbol{k} = \left(1 \mid x_1 \quad x_2 \mid x_1^2 \quad x_1x_2 \quad x_2^2 \mid x_1^3 \quad x_1^2x_2 \quad x_1x_2^2 \quad x_2^3 \right)^T,$$

and a shift function $g(x_1, x_2, x_3) = 3x_1^2 + 2x_2^2$. We can write

$$\begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \left(3x_1^2 + 2x_2^2\right) = \begin{pmatrix} 3x_1^2 + 2x_2^2 \\ 3x_1^3 + 2x_1x_2^2 \\ 3x_1^2x_2 + 2x_2^3 \end{pmatrix},$$

which can alternatively be expressed as $\boldsymbol{S}_1 \boldsymbol{k} g(\boldsymbol{x}) = \boldsymbol{S}_g \boldsymbol{k}$, with

$$\begin{aligned} \boldsymbol{S}_1 &= \left(\begin{array}{c|cc|ccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \quad \text{and} \\ \boldsymbol{S}_g &= \left(\begin{array}{c|cc|ccc|cccc} 0 & 0 & 0 & 3 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 2 \end{array}\right). \end{aligned}$$

The above shift property in the monomial basis vectors can be applied to the multivariate Vandermonde basis $\boldsymbol{K}$ at once by means of the introduction of $\boldsymbol{D}_g := \operatorname{diag}\left(g(\boldsymbol{x}^{(1)}), g(\boldsymbol{x}^{(2)}), \ldots g(\boldsymbol{x}^{(m_B)})\right)$. This leads to

$$\boldsymbol{S}_1 \boldsymbol{K} \boldsymbol{D}_g = \boldsymbol{S}_g \boldsymbol{K}, \tag{6.1}$$

which is an eigenvalue problem. In the following paragraphs, we will establish how the shift property together with a numerical basis for the null space leads to an eigenvalue problem from which the solutions of the system (5.1) are found.

Let us consider an example.

Example 6.5. Consider the equations from Example 5.13 with the solutions $(0, 0)$ and $(2, 4)$:

$$\begin{array}{rcccl} f_1(x_1, x_2) & = & x_2 - x_1^2 & = & 0, \\ f_2(x_1, x_2) & = & x_2 - 2x_1 & = & 0, \end{array}$$

We set $g(x_1, x_2) = 2x_1 + 3x_2$. We can now write

$$\boldsymbol{S}_1 \boldsymbol{K}(2) \boldsymbol{D}_g = \boldsymbol{S}_g \boldsymbol{K}(2),$$

or

$$\left(\begin{array}{c|cc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right) \left(\begin{array}{c|c} 1 & 1 \\ \hline 0 & 2 \\ 0 & 4 \\ \hline 0 & 4 \\ 0 & 8 \\ 0 & 16 \end{array}\right) \left(\begin{array}{c|c} 0 & \\ \hline & 16 \end{array}\right) = \left(\begin{array}{c|cc|ccc} 0 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 \end{array}\right) \left(\begin{array}{c|c} 1 & 1 \\ \hline 0 & 2 \\ 0 & 4 \\ \hline 0 & 4 \\ 0 & 8 \\ 0 & 16 \end{array}\right),$$

where in $\boldsymbol{D}_g$ the evaluation of g at $(0, 0)$ and $(2, 4)$ can be recognized, *i.e.*, $g(0, 0) = 0$ and $g(2, 4) = 16$.

The following example shows that one needs to take precaution with applying the shift relation in the case of roots with multiplicity. Instead of using the diagonal matrix $\boldsymbol{D}_g$, a matrix $\boldsymbol{J}_g$ exhibiting a Jordan-like structure will be required.

Example 6.6. We revisit Example 5.24. We have

$$\begin{array}{lclclcl} f_1(x_1,x_2) & = & (x_2-2)^2 & = & x_2^2-4x_2+4 & = & 0, \\ f_2(x_1,x_2) & = & (x_1-x_2+1)^2 & = & x_1^2-2x_1x_2+2x_1+x_2^2-2x_2+1 & = & 0, \end{array}$$

with a single solution $(1,2)$ with multiplicity 4. It is easy to see that the shift relation $\boldsymbol{S}_1\boldsymbol{K}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{K}$ does not exactly hold in this case. Indeed, it is easy to verify that we need to replace the diagonal matrix $\boldsymbol{D}_g$ by a generalization of the Jordan matrix. We have for the x_1 shift

$$\boldsymbol{S}_1\begin{pmatrix} | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ | & | & | & | \end{pmatrix}\boldsymbol{J}_{x_1} = \boldsymbol{S}_{x_1}\begin{pmatrix} | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ | & | & | & | \end{pmatrix},$$

or

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ x_1 & 1 & 0 & 0 \\ x_2 & 0 & 1 & 0 \\ x_1^2 & 2x_1 & 0 & 2 \\ x_1x_2 & x_2 & x_1 & 1 \\ x_2^2 & 0 & 2x_2 & 0 \end{pmatrix}\begin{pmatrix} x_1 & 1 & 0 & 0 \\ & x_1 & 0 & 2 \\ & & x_2 & 1 \\ & & & x_1 \end{pmatrix} = \begin{pmatrix} x_1 & 1 & 0 & 0 \\ x_1^2 & 2x_1 & 0 & 2 \\ x_1x_2 & x_2 & x_1 & 1 \\ x_1^3 & 3x_1^2 & 0 & 6x_1 \\ x_1^2x_2 & 2x_1x_2 & x_1^2 & 2x_2+2x_1 \\ x_1x_2^2 & x_2^2 & 2x_1x_2 & 2x_2 \end{pmatrix}.$$

For a shift with x_2 we have

$$\boldsymbol{S}_1\begin{pmatrix} | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ | & | & | & | \end{pmatrix}\boldsymbol{J}_{x_2} = \boldsymbol{S}_{x_2}\begin{pmatrix} | & | & | & | \\ \partial_{00} & \partial_{10} & \partial_{01} & 2\partial_{20}+\partial_{11} \\ | & | & | & | \end{pmatrix},$$

or

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ x_1 & 1 & 0 & 0 \\ x_2 & 0 & 1 & 0 \\ x_1^2 & 2x_1 & 0 & 2 \\ x_1x_2 & x_2 & x_1 & 1 \\ x_2^2 & 0 & 2x_2 & 0 \end{pmatrix}\begin{pmatrix} x_2 & 0 & 1 & 0 \\ & x_2 & 0 & 1 \\ & & x_2 & 0 \\ & & & x_2 \end{pmatrix} = \begin{pmatrix} x_2 & 0 & 1 & 0 \\ x_1x_2 & x_2 & x_1 & 1 \\ x_2^2 & 0 & 2x_2 & 0 \\ x_1^2x_2 & 2x_1x_2 & x_1^2 & 2x_2+2x_1 \\ x_1x_2^2 & x_2^2 & 2x_1x_2 & 2x_2 \\ x_2^3 & 0 & 3x_2^2 & 0 \end{pmatrix}.$$

We see that the diagonal of $\boldsymbol{J}$ contains the shifts, which means that from the eigenvalues of $\boldsymbol{S}_1\boldsymbol{K}\boldsymbol{J} = \boldsymbol{S}_g\boldsymbol{J}$ we can obtain the x_1 and x_2 components, but due to the multiplicity of the eigenvalues, the reconstruction of $\boldsymbol{K}$ will not work.

6.2 Null Space Based Root-finding

6.2.1 Generic Case

After the construction of the suitably sized Macaulay matrix $\boldsymbol{M} := \boldsymbol{M}(d_G)$ having nullity$(\boldsymbol{M}) = m_B$, the multivariate Vandermonde null space $\boldsymbol{K}$ is not available directly, but instead a basis for the null space of $\boldsymbol{M}$ can be computed as $\boldsymbol{Z}$. A numerically reliable way to obtain $\boldsymbol{Z}$ is by means of a singular value decomposition (Golub and Van Loan, 1996). Observe now that $\boldsymbol{K}$ can be written as $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$ where $\boldsymbol{T}$ is non-singular. In combination with the previous results this brings us to the first main theorem which phrases the root-finding problem as an eigenvalue problem.

Theorem 6.7 (Null space based root-finding)**.** Assume that the system (5.1) has only affine roots and consider a shift polynomial $g(\boldsymbol{x})$. Let $\boldsymbol{S}_1$ select the rows from $\boldsymbol{Z}$ corresponding to the m_B standard monomials, and let $\boldsymbol{S}_g$ select their corresponding mappings after multiplication with the shift function $g(\boldsymbol{x})$. The eigenvalues of the generalized eigenvalue problem

$$\boldsymbol{S}_1\boldsymbol{Z}\left(\boldsymbol{T}\boldsymbol{D}_g\boldsymbol{T}^{-1}\right) = \boldsymbol{S}_g\boldsymbol{Z}, \tag{6.2}$$

are the evaluations of the roots of (5.1) at the shift function $g(\boldsymbol{x})$. The matching between the individual components x_i can be obtained by computing $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$ and then rescaling the columns of $\boldsymbol{K}$ such that the first row entries equal one.

6.2.2 Two Ways To Use the Shift Property

There are essentially two ways to phrase the eigenvalue problem. The difference lies in which rows of $\boldsymbol{Z}$ are selected by $\boldsymbol{S}_1$.

1. The first way is to obtain the regular square eigenvalue problem by letting $\boldsymbol{S}_1$ select the first m_B linearly independent rows of $\boldsymbol{Z}$.
2. Secondly, it is possible to let $\boldsymbol{S}_1$ select all degree-blocks of $\boldsymbol{Z}$ which have degrees 0 up to $d^\star - 1$, resulting in a rectangular generalized eigenvalue problem.

The resulting generalized eigenvalue problem $\boldsymbol{S}_1\boldsymbol{Z}\left(\boldsymbol{T}\boldsymbol{D}_g\boldsymbol{T}^{-1}\right) = \boldsymbol{S}_g\boldsymbol{Z}$ is either square or rectangular, but can always be turned into a square eigenvalue problem by considering the pseudo-inverse of $\boldsymbol{S}_1\boldsymbol{Z}$, where $\boldsymbol{Z}$ is of full column rank. We have in both cases

$$\left(\boldsymbol{S}_1\boldsymbol{Z}\right)^+ \boldsymbol{S}_g\boldsymbol{Z} = \boldsymbol{T}\boldsymbol{D}_g\boldsymbol{T}^{-1}.$$

If (6.2) has multiple eigenvalues, the evaluation of $g(\boldsymbol{x})$ at the roots correctly corresponds to the eigenvalues, but the individual components x_i cannot be reconstructed from $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$. Mutual matching of the solution components can in such cases only be achieved by solving the eigenvalue problem for consecutive shift functions $g(\boldsymbol{x}) = x_i$, for $i = 1, \ldots, n$, and exhaustively combining the results by evaluating them in the system (5.1).

Example 6.8. We will illustrate the null space based root-finding procedure for the generic case by means of an example. Consider the equations

$$\begin{array}{rclcl} f_1(x_1,x_2,x_3) & = & x_1^2+5x_1x_2+4x_2x_3-10 & = & 0, \\ f_2(x_1,x_2,x_3) & = & x_2^3+3x_1^2x_2-12 & = & 0, \\ f_3(x_1,x_2,x_3) & = & x_3^3+4x_1x_2x_3-8 & = & 0, \end{array}$$

where $d_1 = 2$ and $d_2 = d_3 = 3$. We denote $d_\circ = \max(d_i) = 3$, and hence we start the Macaulay matrix construction at degree $d = 3$. As equation f_1 is of degree 2, we first adjoin the shifted versions $x_1 f_1$, $x_2 f_1$ and $x_3 f_1$ to the matrix $\boldsymbol{M}(3)$ so that we generate a maximum number of polynomials of degree 3. This gives rise to the matrix $\boldsymbol{M}(3)$ as

$$\left(\begin{array}{r|rrr|rrrrrr|rrrrrrrrrr} -10 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 5 & 0 & 0 & 0 & 4 & 0 \\ -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 1 \end{array}\right),$$

of which the rows correspond to the equations f_1, $x_1 f_1$, $x_2 f_1$, $x_3 f_1$, f_2 and f_3 and the columns to the monomials 1, x_1, x_2, x_3, x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2, x_1^3, $x_1^2x_2$, $x_1^2x_3$, $x_1x_2^2$, $x_1x_2x_3$, $x_1x_3^2$, x_2^3, $x_2^2x_3$, $x_2x_3^2$, x_3^3.

The Macaulay matrix becomes too large to print for higher degrees, so we have summarized the matrix sizes, nullities and the indices of the linearly independent monomials of $\boldsymbol{M}(d)$ (in the degree negative lexicographic ordering) for the consecutive degrees d in Table 6.1.

Table 6.1: Diagram for Example 6.8 showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of degree d. We print the size of the Macaulay matrix, its nullity and the indices (counted in the degree negative lexicographic order) of the linearly independent monomials in the null space of the Macaulay matrix. The nullity stabilizes at the value $m_B = 18$ at degree $d = 5$. At the same degree the linear independent monomials stabilize.

d	size $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	standard monomials (index)
3	6×20	14	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16
4	18×35	17	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 21, 22, 23
5	40×56	18	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 21, 22, 23, 36
$d^\star = 6$	75×84	18	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 21, 22, 23, 36
7	126×120	18	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 21, 22, 23, 36
8	196×165	18	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 16, 21, 22, 23, 36

The number of rows $p(d)$ and the number of columns $q(d)$ of $\boldsymbol{M}(d)$ is given by

$$\begin{aligned} p(d) &= \binom{d+1}{d-2} + 2\binom{d}{d-3} &&= \tfrac{(d+1)!}{2!\cdot(d-1)!} + 2\tfrac{d!}{3!\cdot(d-3)!}, \\ & &&= \tfrac{1}{3}d^3 - d^2 + \tfrac{1}{2}d, \\ q(d) &= \binom{d+3}{d} &&= \tfrac{(d+3)!}{3!\cdot d!}, \\ & &&= \tfrac{1}{6}d^3 + d^2 + \tfrac{11}{6}d + 1. \end{aligned}$$

Note that there are in this example 18 monomials that 'stabilize', as predicted by the Bézout number: $m_B = 2 \cdot 3 \cdot 3 = 18$.

We consider the Macaulay matrix for degree $d^\star = 6$. As in the previous example, we can now compute a basis for the null space of $M(6)$ as $\boldsymbol{Z}$. We set up the generalized eigenvalue problems as in (6.2) using a random shift function $g(x_1, x_2, x_3)$ from which we correctly retrieve the 18 solutions:

x_1	x_2	x_3
$0.3404 \pm 0.1844i$	$-1.1743 \pm 2.0411i$	$-0.9622 \mp 1.1366i$
$0.3710 \pm 0.7693i$	$2.4894 \mp 0.2483i$	$0.5815 \mp 0.9147i$
$2.1409 \mp 1.5774i$	$0.1642 \pm 0.5484i$	$1.1399 \mp 0.4895i$
$-0.9055 \pm 0.6846i$	$-1.5288 \mp 2.3832i$	$0.5798 \mp 0.1978i$
$1.7601 \mp 1.8284i$	$-0.0356 \pm 0.6180i$	$0.1569 \mp 1.9950i$
-1.5022	1.3823	3.2782
$2.8552 \pm 0.8711i$	$0.3748 \mp 0.2482i$	$-0.8328 \mp 2.5953i$
$-0.0336 \pm 1.8709i$	$-1.3900 \pm 0.1114i$	$-2.3686 \mp 2.5545i$
$-4.0364 \pm 0.1109i$	$0.2447 \pm 0.0134i$	$-1.3030 \pm 1.1237i$
-3.4818	0.3290	2.7391

6.2.3 About the Choice of Basis

The 'multiplicative shift structure' is indeed a property of the null space as a vector space, and not of the specific choice of basis. This was used in phrasing the generalized eigenvalue problem in any arbitrary basis for the null space, such as for instance a basis for the null space $\boldsymbol{Z}$ obtained using SVD (which is the preferred basis from the numerical point of view).

Let us now look at what the repercussions are of choosing a particular basis for the null space of $\boldsymbol{M}$. We consider the following three bases for phrasing the eigenvalue problem:

1. For the Vandermonde basis $\boldsymbol{K}$ for the null space of $\boldsymbol{M}$, containing the evaluation of the roots of the system in the multivariate Vandermonde

vector, the eigenvalue problem becomes

$$(S_1 \boldsymbol{K})\boldsymbol{I}\boldsymbol{D}_g = (S_g \boldsymbol{K})\boldsymbol{I},$$

from which we see that the eigenvector matrix is $\boldsymbol{I}$.

2. Consider the canonical basis for the null space $\boldsymbol{H} = \boldsymbol{K}\boldsymbol{V}$, defined as in Chapter 3, where $\boldsymbol{V}$ is composed of vectors containing the standard monomials evaluated at each of the solutions $\boldsymbol{x}^{(i)}$. The eigenvalue problem in the canonical basis is

$$(S_1 \boldsymbol{H})\boldsymbol{V}\boldsymbol{D}_g = (S_g \boldsymbol{H})\boldsymbol{V},$$

where $S_1 \boldsymbol{H} = \boldsymbol{I}$ and the eigenvectors have the Vandermonde structure (in the standard monomials).

3. Let $\boldsymbol{Z}$ be a numerically computed basis for the null space, defined by $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$. The eigenvalue problem in the numerical basis is

$$(S_1 \boldsymbol{Z})\boldsymbol{T}\boldsymbol{D}_g = (S_g \boldsymbol{Z})\boldsymbol{T},$$

having the same eigenvalues (*i.e.,* $\boldsymbol{D}_g$) as in the other bases. There is no specific structure present in $\boldsymbol{T}$ in this case, but by computing $\boldsymbol{Z}\boldsymbol{T}$ the reconstruction of the multivariate Vandermonde null space is possible.

6.2.4 Solutions at Infinity

As we have discussed in Section 5.3.3, there are two ways to deal with the solutions at infinity. Either one removes from the Macaulay coefficient matrix $\boldsymbol{M}$ the columns corresponding to the standard monomials associated with the solutions at infinity, leading to the reduced Macaulay matrix $\boldsymbol{M}^\star$, which can then be used immediately to phrase the generalized eigenvalue problem (6.2). Alternatively, the phenomenon by which we have observed that the standard monomials associated to the solutions at infinity move along to higher degrees as the degree d increases, can be used to separate in the null space of $\boldsymbol{M}(d)$ the columns corresponding to the affine roots and the columns corresponding to the solutions at infinity. In this case, the set of affine standard monomials $B^\star(d_G)$ as established in Definition 5.21 is considered.

In the current section we will develop the latter way to deal with the solutions at infinity, which we will achieve by performing a column compression on a numerical basis for the null space $\boldsymbol{Z}$. Since $\boldsymbol{Z}$ by definition contains both null space vectors generated by affine solutions and null space vectors generated by solutions at infinity, $\boldsymbol{Z}$ needs to be altered as to obtain $\boldsymbol{W}$ having m_a columns corresponding to the solutions at infinity only. This is achieved by the column compression of $\boldsymbol{Z}$, which is defined as follows.

Theorem 6.9 (Column compression)**.** Consider $\boldsymbol{Z}$ of size $q \times m$ and define

$$\boldsymbol{Z} := \begin{array}{c c} & m \\ \begin{array}{c} k \\ q-k \end{array} & \begin{pmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \end{pmatrix} \end{array},$$

with $\mathrm{rank}(\boldsymbol{Z}_1) = m_a < m$. The SVD of $\boldsymbol{Z}_1$ is given by $\boldsymbol{Z}_1 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{Q}^T$. Then $\boldsymbol{W} := \boldsymbol{Z}\boldsymbol{Q}$ is called the *column compression* of $\boldsymbol{Z}$ and can be partitioned as

$$\boldsymbol{W} = \begin{array}{c c} & \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{array}{c} k \\ q-k \end{array} & \begin{pmatrix} \boldsymbol{W}_{11} & \boldsymbol{0} \\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} \end{pmatrix} \end{array}. \tag{6.3}$$

Proof. The partitioning of the SVD of $\boldsymbol{Z}_1$ is given by

$$\boldsymbol{Z}_1 = \begin{array}{c c} & \begin{array}{cc} m_a & k-m_a \end{array} \\ k & \begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{\Sigma}_{m_a} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \end{array} \begin{array}{c} m \\ \begin{pmatrix} \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_2^T \end{pmatrix} \end{array}.$$

Then $\boldsymbol{W} := \boldsymbol{Z}\boldsymbol{Q}$ immediately leads to

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{Z}_1\boldsymbol{Q}_1 & \boldsymbol{Z}_1\boldsymbol{Q}_2 \\ \boldsymbol{Z}_2\boldsymbol{Q}_1 & \boldsymbol{Z}_2\boldsymbol{Q}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{Z}_1\boldsymbol{Q}_1 & \boldsymbol{0} \\ \boldsymbol{Z}_2\boldsymbol{Q}_1 & \boldsymbol{Z}_2\boldsymbol{Q}_2 \end{pmatrix}.$$

□

The root-finding method described in Section 6.2.1 can now be used on a certain block of $\boldsymbol{W}$ to phrase an eigenvalue problem from which we can find the affine roots of (5.1).

Said in other words, we want to work in the 'above the gap' part of $\boldsymbol{Z}$, in which interference with columns corresponding to the roots at infinity is eliminated. This is achieved by means of the column compression technique.

Theorem 6.10 (Null space based polynomial system solving)**.** Let $\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{Q}$ denote the column compression of $\boldsymbol{Z}$ as in Theorem 6.9, and let $\boldsymbol{W}_{11}$ denote the north-western block of $\boldsymbol{W}$ consisting of the first k rows and the first m_a columns (as in (6.3)). The evaluation of the shift function $g(\boldsymbol{x})$ at the m_a affine roots are then the eigenvalues of the generalized eigenvalue problem

$$\boldsymbol{S}_1\boldsymbol{W}_{11}\boldsymbol{V}_{11}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{W}_{11}\boldsymbol{V}_{11},$$

where $\boldsymbol{S}_1$, $\boldsymbol{D}_g$ and $\boldsymbol{S}_g$ are defined in agreement with Proposition 6.3, and $\boldsymbol{V}_{11}$ is the $m_a \times m_a$ north-western block of $\boldsymbol{V} = \boldsymbol{Q}^{-1}\boldsymbol{T}$.

Proof. Proposition 6.3 holds for the generic case in which there are only affine roots. When there are solutions at infinity, some vectors in $\boldsymbol{K}$ are caused by the solutions at infinity. Let us denote by m_a the number of affine roots, and let $\boldsymbol{K}_a$ (size $q \times m_a$) contain the Vandermonde structured basis vectors evaluated at the m_a affine roots. We will now work towards an expression of the form

$$\boldsymbol{S}_1\boldsymbol{K}_a\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{K}_a.$$

Since $\boldsymbol{K}_a$ cannot be obtained directly, we work with a numerical basis for the null space of $\boldsymbol{M}$, which we denote by $\boldsymbol{Z}$. We have that $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$, with $\boldsymbol{T}$ a nonsingular matrix expressing a linear change of basis.

In order to separate the affine roots from the solutions at infinity in $\boldsymbol{Z}$, we proceed as follows. Let

$$\boldsymbol{Z} := \begin{array}{c} \\ k \\ q-k \end{array} \begin{array}{c} m \\ \begin{pmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \end{pmatrix} \end{array},$$

where $\operatorname{rank} \boldsymbol{Z}_1 = m_a$. (The choice of the number of rows k will be elaborated further on). The Vandermonde structured basis for the null space $\boldsymbol{K}$ is partitioned accordingly as

$$\boldsymbol{K} = \begin{array}{c} \\ k \\ q-k \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{K}_{a1} & \boldsymbol{K}_{\infty 1} \\ \boldsymbol{K}_{a2} & \boldsymbol{K}_{\infty 2} \end{pmatrix} \end{array}.$$

We now perform a column compression on $\boldsymbol{Z}$ as to obtain $\boldsymbol{W}$ as in Theorem 6.9. Since $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T}$ and $\boldsymbol{W} = \boldsymbol{Z}\boldsymbol{Q}$, there exists a nonsingular $\boldsymbol{V} := \boldsymbol{Q}^{-1}\boldsymbol{T}$ such that we can express $\boldsymbol{K}$ as

$$\begin{array}{rcl} \boldsymbol{K} & = & \boldsymbol{Z}\boldsymbol{T}, \\ & = & (\boldsymbol{Z}\boldsymbol{Q})(\boldsymbol{Q}^{-1}\boldsymbol{T}), \\ & = & \boldsymbol{W}\boldsymbol{V}. \end{array}$$

Hence,

$$\begin{array}{rcl} \boldsymbol{K} & = & \begin{array}{c} \\ k \\ q-k \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{W}_{11} & \boldsymbol{0} \\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{V}_{11} & \boldsymbol{V}_{12} \\ \boldsymbol{V}_{21} & \boldsymbol{V}_{22} \end{pmatrix} \end{array}, \\ & = & \begin{array}{c} \\ k \\ q-k \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{W}_{11}\boldsymbol{V}_{11} & \boldsymbol{W}_{11}\boldsymbol{V}_{12} \\ \boldsymbol{W}_{21}\boldsymbol{V}_{1}1 + \boldsymbol{W}_{22}\boldsymbol{V}_{2}1 & \boldsymbol{W}_{21}\boldsymbol{V}_{12} + \boldsymbol{W}_{22}\boldsymbol{V}_{22} \end{pmatrix} \end{array}, \\ & = & \begin{array}{c} \\ k \\ q-k \end{array} \begin{array}{c} \begin{array}{cc} m_a & m-m_a \end{array} \\ \begin{pmatrix} \boldsymbol{K}_{a1} & \boldsymbol{0} \\ \boldsymbol{K}_{a2} & \boldsymbol{K}_{\infty 2} \end{pmatrix} \end{array}. \end{array}$$

From $\boldsymbol{K}_{a1} = \boldsymbol{W}_{11}\boldsymbol{V}_{11}$ the choice for k is done such that the shift property $\boldsymbol{S}_1\boldsymbol{K}_{a1}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{K}_{a1}$ can be written. Finally, by replacing $\boldsymbol{K}_{a1}$ by $\boldsymbol{W}_{11}\boldsymbol{V}_{11}$ we find the generalized eigenvalue problem in terms of the numerically computed basis for the null space of $\boldsymbol{M}$ (with the column compression) as

$$\boldsymbol{S}_1\boldsymbol{W}_{11}\boldsymbol{V}_{11}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{W}_{11}\boldsymbol{V}_{11}.$$

□

Corollary 6.11. The mutual matching between the solution components can be recovered by computing $\boldsymbol{K}_{a1} = \boldsymbol{W}_{11}\boldsymbol{V}_{11}$ and rescaling the result column-wise such that the first row contains ones.

The following corollary states that it is never necessary to explicitly determine the set of affine standard monomials; it suffices to do rank-tests to determine at which degree the gap occurs between the affine standard monomials and the standard monomials of the roots at infinity.

Corollary 6.12. It is not necessary to explicitly determine the set of affine standard monomials $B^\star$. Instead, d_G can be defined as the degree for which the rank does not change by adding an additional degree block in the null space, *i.e.,* there are no 'new' linearly independent monomials found. Consequently, by letting $\boldsymbol{S}_1$ select *all* low degree monomials (ensuring that the affine standard monomials are included), the rectangular eigenvalue problem will provide the roots.

Due to the dual rank property between rows in the null space and the columns of the Macaulay matrix, one can alternatively scan over the columns of $\boldsymbol{M}$ (per degree-block) and monitor the rank increases, thereby going from right to left.

Corollary 6.12 is quite important since a critical step in the classical computer algebra root-finding methods is determining the affine standard monomials (Stetter, 2004, Chapter 10); making a wrong choice will inevitably lead to numerical instabilities. In our approach the risk of choosing a 'wrong' monomial is impossible, as all monomials are considered.

The complete null space based root-finding procedure is summarized in Algorithm 3.

Algorithm 3. *(Affine null space based root-finding)*

```
input:   system of $n$ equations $f_i = 0$ having
         total degrees $d_i$ in $n$ unknowns $x_i$
output:  $m_a$ affine evaluations of the solutions
         at the function $g(\boldsymbol{x})$

  1. Determine $B^\star(d_G)$, and let $m_a = \#B^\star(d_G)$ (Corollary 6.12)
```

2. Let $\boldsymbol{M} := \boldsymbol{M}(d_G)$ and let $\boldsymbol{Z} := \boldsymbol{Z}(d_G)$
3. Column compression of $\boldsymbol{Z}$ yields $\boldsymbol{W}_{11}$ (Theorem 6.9)
   (Note that if $m_a = m_B$, $\boldsymbol{W}_{11} := \boldsymbol{Z}$)
4. The m_a affine roots evaluated at a user-defined shift function $g(\boldsymbol{x})$ are the eigenvalues of
   $$\boldsymbol{S}_1\boldsymbol{W}_{11}\boldsymbol{V}_{11}\boldsymbol{D}_g = \boldsymbol{S}_g\boldsymbol{W}_{11}\boldsymbol{V}_{11},$$
   where $\boldsymbol{S}_1$, $\boldsymbol{D}_g$ and $\boldsymbol{S}_g$ are defined in accordance with Proposition 6.3.
5. Reconstruct the mutual matching between the solution components x_i from
   $$\boldsymbol{K}_{a1} = \boldsymbol{W}_{11}\boldsymbol{V}_{11},$$
   and a consecutive column rescaling

Example 6.13. Consider the polynomial system

$$\begin{array}{rcl} f_1(x_1,x_2,x_3) & = & x_1x_2 - 3 = 0 \\ f_2(x_1,x_2,x_3) & = & x_1^2 - x_3^2 + x_1x_3 - 5 = 0 \\ f_3(x_1,x_2,x_3) & = & x_3^3 - 2x_1x_2 + 7 = 0, \end{array}$$

with $d_1 = 2$, $d_2 = 2$ and $d_3 = 3$. The Bézout number is $m_B = 2 \cdot 2 \cdot 3 = 12$. We show in Table 6.2 the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d.

Table 6.2: Diagram showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d. The nullity of $\boldsymbol{M}(d)$ stabilizes at the value $m_B = 12$. At degree $d_G = 7$ a degree-gap of one degree has arisen between the standard monomials of the affine solutions and the solutions at infinity. The affine standard monomials are denoted in bold-face.

d	size $\boldsymbol{M}(d)$	nullity $\boldsymbol{M}(d)$	(**affine**) standard monomials
3	9×20	11	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2x_3, x_1^3, x_2^3, x_2^2x_3$
4	24×35	12	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_1^3, x_2^3, x_2^2x_3, x_2^4, x_2^3x_3$
5	50×56	12	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^2, x_2^3, x_2^4, x_2^3x_3, x_2^5, x_2^4x_3$
6	90×84	12	$1, x_1, x_2, x_3, x_1^2, x_1x_3, x_2^3, x_2^4, x_2^5, x_2^4x_3, x_2^6, x_2^5x_3$
$7 =: d_G$	147×120	12	$\mathbf{1}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_1^2}, \mathbf{x_1x_3}, x_2^4, x_2^5, x_2^6, x_2^5x_3, x_2^7, x_2^6x_3$
8	224×165	12	$\mathbf{1}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_1^2}, \mathbf{x_1x_3}, x_2^5, x_2^6, x_2^7, x_2^6x_3, x_2^8, x_2^7x_3$

Inspection of the ranks gives us at d_G the set of affine standard monomials as

$$B^\star(7) = \{1, x_1, x_2, x_3, x_1^2, x_1x_3\}.$$

We construct the Macaulay matrix $\boldsymbol{M}(6)$ having size 90×84 (Figure 5.2 and Figure 5.3) and find a basis for the null space as $\boldsymbol{Z}$ having size 84×12.

The remaining standard monomials are

$$B(d_G)\backslash B^\star(d_G) = \{x_2^4, x_2^5, x_2^6, x_2^5x_3, x_2^7, x_2^6x_3\}.$$

We take $k = 20$, which corresponds to the number of monomials of degrees zero up to three in three variables. Using $m_a = 6$ and $k = 20$ we perform a column compression on $\boldsymbol{Z}$ to find $\boldsymbol{W}_{11}$ with size 20 × 6.

In order to avoid a multiplicity of the evaluation of the roots by the user-defined function g, it is advised to employ a function with random (complex) coefficients. Here we choose as a shift function $g(x_1, x_2, x_3) = x_1 + 2x_2 + 3x_3$ to allow the reader to easily follow the steps of the algorithm.

We determine the selection matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_g$ according to Proposition 6.3. The row selection matrix $\boldsymbol{S}_1$ selects all rows of $\boldsymbol{W}_{11}$ corresponding to the monomials of degrees zero up to two (*i.e.,* the first ten rows). We hence find

$$\boldsymbol{S}_1 = \left(\begin{array}{c|ccc|cccccc|cccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right),$$

and

$$\boldsymbol{S}_g = \left(\begin{array}{c|ccc|cccccc|cccccccccc}
0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 3
\end{array}\right).$$

From the (rectangular) generalized eigenvalue problem

$$\boldsymbol{S}_1 \boldsymbol{W}_{11} \boldsymbol{V}_{11} \boldsymbol{D}_g = \boldsymbol{S}_g \boldsymbol{W}_{11} \boldsymbol{V}_{11},$$

we find the six affine solutions $\boldsymbol{x}^{(i)}, i = 1, \ldots, m_a$ evaluated at the function $g(\boldsymbol{x})$. From $\boldsymbol{K}_{a1} = \boldsymbol{W}_{11}\boldsymbol{V}_{11}$ we correctly reconstruct their mutual matching as in the following table.

x_1	x_2	x_3
$1.857 \mp 0.176i$	$1.600 \pm 0.151i$	$0.500 \pm 0.866i$
-2.000	-1.500	-1.000
$-2.357 \pm 0.689i$	$-1.172 \mp 0.343i$	$0.500 \mp 0.866i$
3.000	1.000	-1.000

The following example shows that, when the affine solution set is zero-dimensional, but the solution set at infinity is positive-dimensional, the root-finding algorithm still works.

Example 6.14. The case that the affine variety is zero-dimensional while the projective variety is one-dimensional may occur. Consider the system

$$\begin{array}{rclcl}
f_1(x_1,x_2,x_3,x_4) & = & x_1+x_2-1 & = & 0, \\
f_2(x_1,x_2,x_3,x_4) & = & x_1x_3+x_2x_4 & = & 0, \\
f_3(x_1,x_2,x_3,x_4) & = & x_1x_3^2+x_2x_4^2-2/3 & = & 0, \\
f_4(x_1,x_2,x_3,x_4) & = & x_1x_3^3+x_2x_4^3 & = & 0.
\end{array}$$

The stabilization diagram is given in Table 6.3. We observe that after a few iterations the nullity keeps increasing with 2. By observing the standard monomials, we observe that the monomials 1 and x_4 stabilize and correspond to two affine solutions. The remaining monomials shift along as the iteration number increases, and moreover, new linear independent monomials are found in each iteration. This indicates that the solution set at infinity is positive-dimensional. The affine roots are retrieved correctly using the algorithms explained above as $(0.5000, 0.5000, 0.8165, -0.8165)$ and $(0.5000, 0.5000, -0.8165, 0.8165)$.

Table 6.3: Stabilization diagram for Example 6.14, showing the properties of the Macaulay matrix $\boldsymbol{M}(d)$ as a function of the degree d.

d	size $\boldsymbol{M}(d)$	rank $M(d)$	nullity $\boldsymbol{M}(d)$
4	56 × 70	50	20
5	125 × 126	103	23
6	246 × 210	185	25
7	441 × 330	303	27
8	736 × 495	466	29
9	1161 × 715	684	31

6.2.5 Iterative Null Space Computations

Exploiting Sparsity and Structure

Due to the rapid dimensional growth of the Macaulay matrix as d increases, the bottleneck of the root-finding algorithm is located at the construction of a numerical basis of the null space of the Macaulay matrix. In the current section we will address this issue and develop an iterative procedure to update the null space as d increases. The algorithm we will develop will exploit the structure of the Macaulay coefficient matrix in order to iteratively compute a basis for the null space.

We will employ the nested Macaulay matrix, denoted by $\boldsymbol{N}(d)$, in order to avoid confusion. The iteratively constructed Macaulay matrix $\boldsymbol{N}(d)$ can be obtained by a row permutation of the Macaulay matrix $\boldsymbol{M}(d)$ (Definition 5.3), as in

$$\boldsymbol{P}\boldsymbol{M}(d) = \boldsymbol{N}(d).$$

A direct consequence is that the (right) null spaces of $\boldsymbol{M}(d)$ and $\boldsymbol{N}(d)$ coincide. The iterative structure can now be exploited in an algorithm to compute a basis for the null space in an iterative way.

Block SVD Motzkin Algorithm

In Chapter 3 we have described the so-called Motzkin algorithm as a naive way to find a numerical basis for the null space of a matrix. This method has its didactical merits, but from a numerical point of view, the procedure is flawed: during the consecutive multiplication of the matrices $\boldsymbol{W}_k$, some elements of $\boldsymbol{b}_k^T \boldsymbol{W}_1 \boldsymbol{W}_2 \cdots \boldsymbol{W}_{k-1}$ may become very small — choosing one of them as a non-zero pivot elements would lead to an incorrect result.

The following procedure addresses the numerical issues by employing SVDs on certain blocks of the (row-reordered) Macaulay matrix. A trick similar to the Motzkin procedure is applied on block-matrix level, instead on matrix row level. We will work with subsequent iterations of the nested quasi-Toeplitz Macaulay matrix $\boldsymbol{N}(d)$.

Let $\boldsymbol{N}(d)$ be the nested quasi-Toeplitz structured Macaulay matrix for degree d. Let $\boldsymbol{Z}(d)$ denote a basis for the null space of $\boldsymbol{N}(d)$ and denote by $p(d)$, $q(d)$ and $c(d)$ the number of rows, the number of columns, and the nullity of $\boldsymbol{N}(d)$, respectively. The nesting property of the Macaulay matrix allows us to write $\boldsymbol{N}(d+1)$ as

$$\boldsymbol{N}(d+1) = \begin{array}{c} \\ p \\ \Delta p \end{array} \begin{array}{c} \begin{array}{cc} q & \Delta q \end{array} \\ \begin{pmatrix} \boldsymbol{N}(d) & \boldsymbol{0} \\ \boldsymbol{N}_1 & \boldsymbol{N}_2 \end{pmatrix} \end{array},$$

where the dimensions are added for clarity and the updates of the number of rows and columns are $\Delta p := p(d+1) - p(d)$ and $\Delta q := q(d+1) - q(d)$. We now have

$$\boldsymbol{N}(d+1)\,\boldsymbol{Z}(d+1) = \begin{array}{c} \\ p \\ \Delta p \end{array} \begin{array}{c} \begin{array}{cc} q & \Delta q \end{array} \\ \begin{pmatrix} \boldsymbol{N}(d) & \boldsymbol{0} \\ \boldsymbol{N}_1 & \boldsymbol{N}_2 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cc} c(d) & \Delta q \end{array} \\ \begin{pmatrix} \boldsymbol{Z}(d) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \end{array} \begin{array}{c} c(d{+}1) \\ \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} \end{array} = \boldsymbol{0},$$

hence,

$$\begin{array}{c} \\ p \\ \Delta p \end{array} \begin{array}{c} \begin{array}{cc} c(d) & \Delta q \end{array} \\ \begin{pmatrix} \boldsymbol{N}(d)\boldsymbol{Z}(d) & \boldsymbol{0} \\ \boldsymbol{N}_1\boldsymbol{Z}(d) & \boldsymbol{N}_2 \end{pmatrix} \end{array} \begin{array}{c} c(d{+}1) \\ \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} \end{array} = \boldsymbol{0},$$

and

$$\begin{array}{cc} & \begin{array}{ccc} c(d) & \Delta q & c(d{+}1) \end{array} \\ \begin{array}{c} p \\ \Delta p \end{array} & \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{N}_1\boldsymbol{Z}(d) & \boldsymbol{N}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} = \mathbf{0}. \end{array}$$

The matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are obtained as a basis for the null space of

$$\begin{pmatrix} \boldsymbol{N}_1\boldsymbol{Z}(d) & \boldsymbol{N}_2 \end{pmatrix}.$$

An update of the basis of the null space $\boldsymbol{Z}$ is hence computed as

$$\boldsymbol{Z}(d+1) = \begin{pmatrix} \boldsymbol{Z}(d)\boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix}.$$

In Algorithm 4 the procedure to iteratively update the numerical basis for the null space of the Macaulay matrix is summarized. Let us end this section with the following important remarks.

Remark 6.15. The updating algorithm provides a way to iteratively compute a basis for the null space of the Macaulay matrix, implying that larger matrices can be processed than when using an SVD. Instead of the size of the full matrix $\boldsymbol{N}(d)$, now the size of $\begin{pmatrix} \boldsymbol{N}_1 & \boldsymbol{N}_2 \end{pmatrix}$ will be the limiting factor in the computation of $\begin{pmatrix} \boldsymbol{X}^T & \boldsymbol{Y}^T \end{pmatrix}^T$, making the root-finding algorithm procedure feasible for larger problems.

Remark 6.16. It is possible to preserve orthogonality of $\boldsymbol{Z}$ if one starts from an orthogonal basis $\boldsymbol{Z}(d_\circ)$ and $\begin{pmatrix} \boldsymbol{X}^T & \boldsymbol{Y}^T \end{pmatrix}^T$ is orthogonal, which is of interest for numerical considerations. This can be obtained by the use of the SVD.

Remark 6.17. Regardless of the procedure to obtain the updates of the null space, the numerical rank determination of $\begin{pmatrix} \boldsymbol{X}^T & \boldsymbol{Y}^T \end{pmatrix}^T$ is a critical step of Algorithm 4. A wrong rank estimation will influence the subsequent iteration steps, resulting in an incorrect dimension of $\boldsymbol{Z}(d)$.

Algorithm 4. *(Iterative null space updating)*

```
input:   system of $n$ equations $f_i$ with
         $d_\circ := \max(d_i)$, requested degree $d$
output:  iteratively constructed Macaulay matrix $\boldsymbol{N}(d)$
         and basis for null space $\boldsymbol{Z}(d)$

  1. Construct the initial Macaulay matrix $\boldsymbol{N}(d_\circ)$
  2. Compute basis for the null space of $\boldsymbol{N}(d_\circ)$ as $\boldsymbol{Z}(d_\circ)$
  3. repeat until $\delta + 1 = d$,
       a) Update of the Macaulay matrix (iterative construction)
          $\boldsymbol{N}(\delta+1) \leftarrow \begin{pmatrix} \boldsymbol{N}(\delta) & \mathbf{0} \\ \boldsymbol{N}_1 & \boldsymbol{N}_2 \end{pmatrix}$
```

```
    b) Compute basis for null space
       $\begin{pmatrix} X \\ Y \end{pmatrix} \leftarrow$ basis for null space of $\begin{pmatrix} N_1 Z(\delta) & N_2 \end{pmatrix}$
    c) Update basis for null space of $N(\delta+1)$ as
       $Z(\delta+1) \leftarrow \begin{pmatrix} Z(\delta)X \\ Y \end{pmatrix}$
done
```

6.3 Column Space Based Root-finding

6.3.1 Generic Case

The method starts from a Macaulay matrix $M := M(d_G)$ with size $p \times q$. We will consider as the null space of M the multivariate Vandermonde basis K of size $q \times m_B$ as defined earlier, where column i of K contains the multivariate Vandermonde monomial vector k evaluated at the i-th root, for all $i = 1, \ldots, m_B$. As it will turn out, the computation of a numerical basis for the null space of M will not be required.

Consider the column reordering of M as $\begin{pmatrix} M_1 & M_2 \end{pmatrix}$, such that

$$\begin{pmatrix} M_1 & M_2 \end{pmatrix} \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} = \mathbf{0}, \tag{6.4}$$

where $\mathrm{rank}(M_2) = \mathrm{rank}(M) = r$ (full column rank) and K_1 is of size $m_B \times m_B$ containing the standard monomial rows of K. Recall that $q - r = m_B$ is the number of solutions. Therefore, M_1 is of size $p \times q - r = p \times m_B$, M_2 is of size $p \times q - m_B$, and K_2 is of size $q - m_B \times m_B$. Now, the shift property (6.1) can be rephrased according to (6.4) where we let S_1 select the standard monomials (*i.e.*, K_1), or

$$\begin{aligned} S_1 K D_g &= S_g K, \\ K_1 D_g &= \begin{pmatrix} \Sigma_1 & \Sigma_2 \end{pmatrix} \begin{pmatrix} K_1 \\ K_2 \end{pmatrix}, \end{aligned} \tag{6.5}$$

in which the row combination matrix S_g is reordered into Σ_1 and Σ_2, hence distinguishing two kinds of 'shifts':

- Some monomials in K_1 will be mapped to monomials in K_1 under the multiplication with $g(x)$. They are represented by the matrix Σ_1.
- The matrix Σ_2 represents shifts of monomials of K_1 that are mapped by the multiplication with $g(x)$ to K_2.

The matrices $\boldsymbol{\Sigma}_1$ (size $m_B \times m_B$) and $\boldsymbol{\Sigma}_2$ (size $m_B \times q - m_B$) are directly obtained in accordance with Proposition 6.3, while respecting the possible reordering of the monomials of $\boldsymbol{M}$ as in (6.4).

Notice that zero columns in $\boldsymbol{\Sigma}_2$ represent the rows of $\boldsymbol{K}_2$ that are not reached by multiplying the standard monomials in $\boldsymbol{K}_1$ with $g(\boldsymbol{x})$. Let us denote by z the number of zero columns in $\boldsymbol{\Sigma}_2$. We find from (6.4) that

$$\boldsymbol{K}_2 = -\boldsymbol{M}_2^{+}\boldsymbol{M}_1\boldsymbol{K}_1,$$

since $\boldsymbol{M}_2$ has full column rank. Observe that only the rows of $\boldsymbol{K}_2$ selected by the nonzero elements of $\boldsymbol{\Sigma}_2$ (reached by shifting the standard monomials) have to be determined. Let us now again reorder the columns of $\boldsymbol{M}$ to obtain

$$\begin{pmatrix} \boldsymbol{M}_{22} & \boldsymbol{M}_{21} & \boldsymbol{M}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{K}_{22} \\ \boldsymbol{K}_{21} \\ \boldsymbol{K}_1 \end{pmatrix} = \boldsymbol{0},$$

where $\boldsymbol{M}_1$ and $\boldsymbol{K}_1$ are defined as above. The matrix $\boldsymbol{K}_2$ is repartitioned into $\boldsymbol{K}_{21}$ and $\boldsymbol{K}_{22}$ such that $\boldsymbol{K}_{21}$ contains the rows of interest, and $\boldsymbol{K}_{22}$ contains the remaining rows of $\boldsymbol{K}_2$. Correspondingly, $\boldsymbol{M}_2$ is repartitioned as $\boldsymbol{M}_{21}$ and $\boldsymbol{M}_{22}$. It turns out that by means of a QR decomposition we can determine the rows of interest of $\boldsymbol{K}_{21}$. Consider the QR of $\begin{pmatrix} \boldsymbol{M}_{22} & \boldsymbol{M}_{21} & \boldsymbol{M}_1 \end{pmatrix}$ as

$$\begin{array}{cccc} & z & q-m_B-z & m_B \\ p & \big(\boldsymbol{M}_{22} & \boldsymbol{M}_{21} & \boldsymbol{M}_1\big) \end{array} =$$

$$\begin{array}{cccc} & z & q-m_B-z & p-q+m_B \\ p & \big(\boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \boldsymbol{Q}_3\ \big) \end{array} \quad \begin{array}{ccc} z & q-m_B-z & m_B \\ \multicolumn{3}{c}{\begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \boldsymbol{R}_{13} \\ & \boldsymbol{R}_{22} & \boldsymbol{R}_{23} \\ & & \boldsymbol{R}_{33} \end{pmatrix}} \end{array}, \qquad (6.6)$$

where the sizes of the matrix blocks are indicated for the sake of clarity. Observe that due to the rank deficiency of $\boldsymbol{M}$ the block $\boldsymbol{R}_{33}$ is either (numerically) zero, or it has zero rows (if $\operatorname{rank}(\boldsymbol{M}) = p$).

We are now ready to phrase the root-finding problem as an eigenvalue problem.

Theorem 6.18 (Column space based root-finding)**.** Assume that the system (5.1) has only affine roots and consider a shift polynomial $g(\boldsymbol{x})$. Let $\boldsymbol{M}_1$, $\boldsymbol{M}_{21}$, $\boldsymbol{M}_{22}$, $\boldsymbol{K}_1$, $\boldsymbol{K}_{21}$, $\boldsymbol{K}_{22}$, and the QR decomposition of $\begin{pmatrix} \boldsymbol{M}_{22} & \boldsymbol{M}_{21} & \boldsymbol{M}_1 \end{pmatrix}$ be defined as above. The root-finding problem is phrased as the eigenvalue problem

$$\boldsymbol{K}_1\boldsymbol{D}_g = \left(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2'\boldsymbol{R}_{22}^{-1}\boldsymbol{R}_{23}\right)\boldsymbol{K}_1, \qquad (6.7)$$

where $\boldsymbol{\Sigma}_2'$ denotes the matrix $\boldsymbol{\Sigma}_2$ with the zero columns removed, having size $m_B \times q - m_B - z$.

Proof. From (6.6) and $\begin{pmatrix} M_{22} & M_{21} & M_1 \end{pmatrix} \begin{pmatrix} K_{22}^T & K_{21}^T & K_1^T \end{pmatrix}^T = \mathbf{0}$ the rows of interest are obtained as $K_{21} = -R_{22}^{-1} R_{23} K_1$. Combining this with (6.5), we obtain (6.7). □

Corollary 6.19. The eigenvectors of (6.7) obey the multivariate monomial structure of K_1. Provided that the components x_i occur as standard monomials (*i.e.,* they are elements of K_1), the mutual matching between the components x_i can therefore be reconstructed by rescaling the eigenvectors such that the entries corresponding to the monomials 1 (typically taken as the first row entries in a graded ordering) are scaled to 1.

6.3.2 Solutions at Infinity

If there are solutions at infinity, the set of affine standard monomials $B^\star(d_G)$ as established in Definition 5.21 must be considered. After removing from $M(d_G)$ the columns corresponding to the standard monomials of the solutions at infinity, one obtains $M^\star(d_G)$ and correspondingly $K^\star(d_G)$, which can be used in Theorem 6.18 to find the affine solutions. The complete column space based algorithm for finding the affine roots is summarized in Algorithm 5.

Algorithm 5. *(Affine column space based root-finding)*

input: a system of n equations f_i having
total degrees d_i in n unknowns x_i

output: the m_a affine solutions for a user-chosen
shift function $g(x)$

1. Determine $B^\star(d_G)$ and set $m_a = \#B^\star(d_G)$
2. Remove from $M := M(d_G)$ the columns corresponding to the solutions at infinity, leading to $M^\star K^\star = \begin{pmatrix} M_1 & M_2 \end{pmatrix} \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} = \mathbf{0}$ where K_1 represents the m_a affine standard monomials
3. Σ_1 and Σ_2 are determined for the user-defined shift function $g(x)$
4. Partition $M^\star$ into $(M_{22}\ M_{21}\ M_1)$, where M_{21} contains the columns of interest (*i.e.,* corresponding to the monomials hit by the nonzero columns of Σ_2) and M_{22} contains the remaining columns
5. Compute the QR decomposition of $(M_{22}\ M_{21}\ M_1)$

$$\begin{pmatrix} M_{22} & M_{21} & M_1 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 & Q_3 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ & R_{22} & R_{23} \\ & & R_{33} \end{pmatrix}$$

6. The m_a roots evaluated at the shift function $g(x)$ are found from

$$K_1 D_g = \left(\Sigma_1 - \Sigma_2' R_{22}^{-1} R_{23} \right) K_1$$

7. The mutual matching between the solution components x_i is reconstructed by computing the eigenvectors of $\Sigma_1 - \Sigma_2' R_{22}^{-1} R_{23}$ and rescaling them such that the entries $\prod_{i=1}^n x_i^0 = 1$ are scaled to 1

Example 6.20. We revisit Example 6.13 and use $d_G = 7$. We remove from $M(7)$ the columns corresponding to the monomials

$$x_2^4, x_2^5, x_2^6, x_2^5 x_3, x_2^7, x_2^6 x_3,$$

as to obtain $M^\star := M^\star(7)$ having size 147×114. We consider the set of affine standard monomials $B^\star = \left\{1, x_1, x_2, x_3, x_1^2, x_1 x_3\right\}$ and choose again as the shift polynomial $g(x_1, x_2, x_3) := x_1 + 2x_2 + 3x_3$. The monomials that are reached by shifting the affine standard monomials $B^\star$ by the monomials occurring in g are:

$$\begin{array}{rcl} B^\star & = & \left\{1, x_1, x_2, x_3, x_1^2, x_1 x_3\right\}, \\ B^\star \cdot x_1 & \rightarrow & \left\{x_1, x_1^2, x_1 x_2, x_1 x_3, x_1^3, x_1^2 x_3\right\}, \\ B^\star \cdot x_2 & \rightarrow & \left\{x_2, x_1 x_2, x_2^2, x_2 x_3, x_1^2 x_2, x_1 x_2 x_3\right\}, \\ B^\star \cdot x_3 & \rightarrow & \left\{x_3, x_1 x_3, x_2 x_3, x_3^2, x_1^2 x_3, x_1 x_3^2\right\}. \end{array}$$

We now partition $M^\star$ as $\left(\begin{array}{ccc} M_{22} & M_{21} & M_1 \end{array} \right)$, where M_1 contains the columns of $M^\star$ corresponding to the monomials in $B^\star$, M_{21} contains the columns of interest, *i.e.,* the columns (outside of $B^\star$) that are reached by shifting the monomials in $B^\star$, namely the monomials $x_1 x_2$, x_2^2, $x_2 x_3$, x_3^2, x_1^3, $x_1^2 x_2$, $x_1^2 x_3$, $x_1 x_2 x_3$ and $x_1 x_3^2$ and, finally, M_{22} contains the remaining columns.

The matrices Σ_1 and Σ_2' are found in accordance with g as

$$\Sigma_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\Sigma_2' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \end{pmatrix}.$$

Finally we compute the eigenvalue decomposition of $\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2' \boldsymbol{R}_{22}^{-1} \boldsymbol{R}_{23}$. Rescaling the eigenvectors such that the first entry equals one reveals the six affine solutions (with absolute errors of the order 10^{-13}:

x_1	x_2	x_3
$1.857 \mp 0.176i$	$1.600 \pm 0.151i$	$0.500 \pm 0.866i$
-2.000	-1.500	-1.000
$-2.357 \pm 0.689i$	$-1.172 \mp 0.343i$	$0.500 \mp 0.866i$
3.000	1.000	-1.000

6.4 Application: Polynomial Optimization Problems

6.4.1 Motivation and Approach

Polynomial optimization problems are optimization problems composed of a polynomial objective criterion that is solved subject to polynomial equality constraints on the decision variables. Optimization problems occur in nearly all engineering applications, and are typically solved using local optimization routines (Nocedal and Wright, 2006).

The Lagrange multipliers method provide the necessary conditions for optimality of a polynomial optimization problem as a system of polynomial equations. Let $J(x_1, \ldots, x_n)$ denote the polynomial objective criterion and $g_i(x_1, \ldots, x_n) = 0, i = 1, \ldots, n_g$ are the constraints. The polynomial optimization problem is often written in the form

$$\begin{array}{lrcl} \underset{x_1,\ldots,x_n}{\text{minimize}} & J(x_1,\ldots,x_n), & & \\ \text{subject to} & g_1(x_1,\ldots,x_n) & = & 0, \\ & & \vdots & \\ & g_1(x_1,\ldots,x_n) & = & 0, \end{array}$$

The Lagrangian is defined as

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_{n_g}) = J(x_1, \ldots, x_n) + \sum_{i=1}^{n_g} \lambda_i g_i(x_1, \ldots, x_n),$$

where the newly introduced parameters λ_i are called the Lagrange multipliers. A set of necessary conditions for optimality is obtained from the set of equations

$$\begin{array}{rcl} \partial L/\partial x_1 & = & 0, \\ & \vdots & \\ \partial L/\partial x_n & = & 0, \\ \partial L/\partial \lambda_1 & = & 0, \\ & \vdots & \\ \partial L/\partial \lambda_{n_g} & = & 0, \end{array}$$

which constitutes a system of $n + n_g$ polynomial equations in $n + n_g$ unknowns.

6.4.2 System Identification

System identification is a discipline in systems and control theory concerned with the question of finding dynamical system models to explain a given behavior (usually described as input-output data). It can be considered as a technology underlying many problems in systems and control, digital signal processing, biomedical engineering, and many more. The classical reference works are Ljung (1999); Pintelon and Schoukens (2012); Söderström and Stoica (1989); Van Overschee and De Moor (1996).

A general system identification problem is given as the so-called *errors-in-variables* set-up shown in Figure 6.1. The undisturbed input signals are denoted u_0 and the undisturbed output signals are denoted y_0. In some cases, the measurements of u_0 and y_0 are subject to measurement noise, leading to the measurements $u = u_0 + n_u$ and $y = y_0 + ny$, respectively. Certain models may require an additional unobservable 'noise' input e in order to explain as well as possible the input-output measurements.

Currently, two major approaches dominating the field of system identification are the *prediction error methods* (Ljung, 1999) and the subspace system identification approach (Van Overschee and De Moor, 1996). The prediction error methods formulate the system identification problem as a nonlinear optimization problem. Such a problem can in general only be solved up to a locally optimal solution is found, depending on the specific choice of the initial point of the optimization routine. The subspace identification methods employ numerical linear algebra tools and hence have a single unique solution. They perform very well in practice (and often the solution obtained from the subspace identification algorithm is used as an initial point to further refine the model using the prediction error approach). However, there is

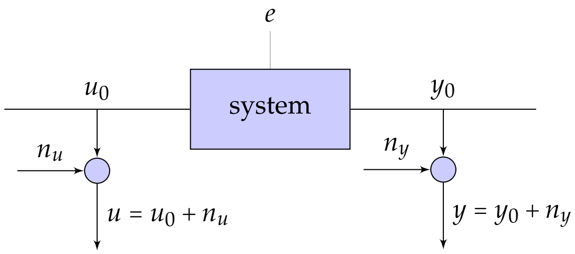

Figure 6.1: Diagram providing a graphical representation of the system identification problem. An unknown system interacts with its environment through the so-called inputs and outputs u_0 and y_0. It is often necessary to suppose that measurement noise n_u and n_y contaminates the measurements of u_0 and y_0, so that only u and y are observable. Additionally, an unobservable noise input e may be used in order to describe the observed input-output behavior.

no explicit cost criterion underlying the subspace identification methods and hence the optimality can be questioned.

The work of Lemmerling and De Moor (2001) proposes a unifying optimization framework where all possible models that can be conceived on the basis of Figure 6.1 are incorporated. This includes several prediction error methods, such as ARMA, ARMAX, as well as 'new' errors-in-variables variations, such as ARMAX with noisy inputs and outputs, dynamical total least squares models, *etc.*

The modeling procedure of Lemmerling and De Moor (2001) reduced to finding a structured low rank approximation of a matrix built from the input-output data, subject to certain model constraints. As it turns out, structured low-rank estimation is a central task in systems theory, identification and control, underlying many modeling problems, ranging from errors-in-variables system identification, over approximate realization theory to model reduction (see Markovsky (2008) for an elaborate survey on the problem and its applications).

Let us consider a simple instance of the structured total least squares problem and apply the null space based method to solve it.

Example 6.21. In this example a 3×3 Hankel structured total least squares problem is solved as a system of polynomial equations to find the globally optimal low-rank approximation to a given data matrix. Let A be a given 3×3 data matrix of full rank, having Hankel matrix structure. De Moor (1993, 1994b) proposes a non-linear generalization of the SVD to solve the STLS problem, which is called the Riemannian SVD, which essentially comprises a system of multivariate polynomial equations. Let $\boldsymbol{v} = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T$ and

$\boldsymbol{l} = \begin{pmatrix} l_1 & l_2 & l_3 \end{pmatrix}^T$. The Riemannian SVD equations are

$$\begin{array}{rcl} \boldsymbol{A}\boldsymbol{v} & = & \boldsymbol{T}_v \boldsymbol{T}_v^T \boldsymbol{l}, \\ \boldsymbol{A}^T \boldsymbol{l} & = & \boldsymbol{T}_l \boldsymbol{T}_l^T \boldsymbol{v}, \\ \boldsymbol{v}^T \boldsymbol{v} & = & 1, \end{array} \tag{6.8}$$

where $\boldsymbol{T}_v$ and $\boldsymbol{T}_l$ capture the required Hankel structure constraint (distinguishing the Riemannian SVD from the SVD) and are defined as

$$\boldsymbol{T}_v = \begin{pmatrix} v_1 & v_2 & v_3 & & \\ & v_1 & v_2 & v_3 & \\ & & v_1 & v_2 & v_3 \end{pmatrix},$$

and $\boldsymbol{T}_l$ is defined similarly (De Moor, 1993, 1994b). The best low-rank approximation of $\boldsymbol{A}$ is reconstructed from the pair of $(\boldsymbol{v}, \boldsymbol{l})$ vectors that minimize the objective criterion

$$J(\boldsymbol{v}) = \boldsymbol{v}^T \boldsymbol{A}^T (\boldsymbol{T}_v \boldsymbol{T}_v^T)^{-1} \boldsymbol{A}\boldsymbol{v},$$

as described in De Moor (1993, 1994b).

Consider a 3 × 3 full-rank Hankel matrix

$$\boldsymbol{A} = \begin{pmatrix} 7 & -2 & 5 \\ -2 & 5 & 6 \\ 5 & 6 & -1 \end{pmatrix},$$

which is approximated by a Hankel matrix $\boldsymbol{B}$ of rank 2.

Replacing the normalization constraint $\boldsymbol{v}^T\boldsymbol{v} = 1$ in (6.8) by $v_1 = 1$ reduces the number of variables by one. The first equation in $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{T}_v\boldsymbol{T}_v^T\boldsymbol{l}$ now does not carry any information anymore since it stems from a derivation with respect to $v_1 = 1$, and hence it can be dropped.

The resulting system is composed of five polynomial equations in five unknowns, where all equations are of degree three. We apply the method as described in Algorithm 3. For $d_G = 11$, the matrix $\boldsymbol{M} = \boldsymbol{M}(d_G)$ has size 6435 × 4368 and is extremely sparse (with only 60489 nonzero elements). A basis for the null space of $\boldsymbol{M}$ is computed, and the root-counting technique reveals there are 39 affine solutions. The solutions are computed and after discarding the complex solutions, the 13 real solutions are retrieved successfully.

The equivalent objective $J(\boldsymbol{v})$ and the (real) values of the critical points are represented graphically in Figure 6.2. The optimal rank-2 Hankel matrix approximation of $\boldsymbol{A}$ is ultimately retrieved as $\boldsymbol{B}$, where

$$\boldsymbol{B} = \begin{pmatrix} 7.6582 & -0.1908 & 3.2120 \\ -0.1908 & 3.2120 & 1.8342 \\ 3.2120 & 1.8342 & 2.4897 \end{pmatrix}.$$

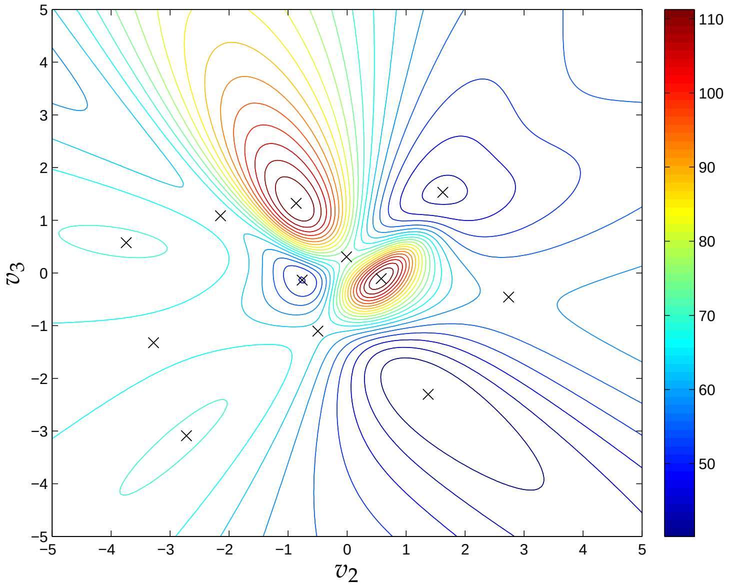

Figure 6.2: Level sets of the minimization problem of the 3 × 3 Hankel structured total least squares problem, given by $J(v) = v^T A^T (T_v T_v^T)^{-1} A v$. The plot shows that the objective has several local optima. The proposed method is able to identify all 13 critical (real) points (12 of which are in the plotted range, indicated by ×), and hence guarantees to retrieve the globally optimal solution.

6.4.3 Power Iterations to Find Minimizing Solution

When one considers a polynomial optimization problem, often one is only interested in the minimizing solution of the objective criterion. In this case, an obvious way to phrase the eigenvalue problem is by using the objective J as a shift criterion. Then we obtain from Proposition 6.3 the eigenvalue problem

$$S_1 Z \left(T D_g T^{-1} \right) = S_g Z,$$

of which the minimizing eigenvalue corresponds to the minimizing solution of the objective function. By using (inverse) power iterations (Golub and Van Loan, 1996) for determining the minimal eigenvalue and eigenvector, the solution of interest can be determined directly.

Polynomial Systems and Realization Theory

7

In Chapter 4 we have alluded to the natural link between polynomial system solving and realization theory. Although the connections between multivariate polynomial system solving and multidimensional realization theory have been described earlier, in these works the notion of a Gröbner basis took a central role. In this chapter, we will explore the links from the perspective of the Macaulay matrix method. It turns out that the null space of the Macaulay matrix can be interpreted as a state sequence of a set of multi-variable difference equations. Conversely, the null space based solution method of Chapter 6 can be interpreted as the application of realization theory on the null space of the Macaulay matrix.

It should be emphasized that this section does not provide any new methods nor algorithms. Rather, it frames the algorithms developed in the previous chapters into the framework of realization theory. Our hope is that the link with the theory of nD systems may provide new insights into polynomial algebra.

7.1 Introduction

The paper by Hanzon and Hazewinkel (2006) lucidly illustrates the natural link between multivariate polynomial system solving and multidimensional systems theory. In simple cases, repeatedly applying the difference equations easily reveals how this can be done. However, in general, it is not clear on beforehand how the state vector needs to be composed and how the equations need to be manipulated in order to arrive to the form where the system can be written in the desired form. As it turns out, the solution is given by constructing a Gröbner basis for the given system of polynomials.

It turns out that the notion of Gröbner bases is not necessary to phrase the eigenvalue problem; instead, the null space based approach boils down to the application of realization theory applied to the null space of the Macaulay matrix, ultimately leading to the eigenvalue formulation such as done in Stetter's method.

There are indications that the solutions at infinity can be described in a similar fashion, however, at this point it is not entirely clear how the system of difference equations should be conceived in order to fully describe this effect. We will illustrate this by means of a simple example, but have observed that the proposed method does not hold in general.

7.2 Generic Systems: Affine Roots Only

Let us start with formalizing the mapping between polynomials and time-shifted signals. In the multivariate (nD) case, the signals have $n > 1$ independent indices as opposed to a single independent index as in the 1D case.

Definition 7.1 (Multivariate Polynomials as Multidimensional Shift Operators)**.** With the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ the nD shift operator $\sigma_1^{\alpha_1} \sigma_2^{\alpha_2} \cdots \sigma_n^{\alpha_n}$ can be associated, acting on a multidimensional signal as

$$\sigma_1^{\alpha_1} \sigma_2^{\alpha_2} \cdots \sigma_n^{\alpha_n} : v(k_1, k_2, \ldots, k_n) \mapsto v(k_1 + \alpha_1, k_2 + \alpha_2, \ldots, k_n + \alpha_n).$$

In the case there are only affine roots, the well-known multivariate Vandermonde structure in the null space of the Macaulay matrix is reminiscent to the natural link with nD realization theory.

The key observation in the interpretation of the polynomial system solving problem as a question in realization theory is the fact that the multivariate Vandermonde structured matrix lies in the null space of the Macaulay matrix. The simplified Attasi model as introduced in Chapter 3 is given as

$$v(k_1, \ldots, k_{i-1}, k_i + 1, k_{i+1}, \ldots, k_n) = \boldsymbol{A}_i v(k_1, \ldots, k_n),$$

for all $i = 1, \ldots, n$. The action matrices $\boldsymbol{A}_i \in \mathbb{R}^{\theta \times \theta}$ form a commuting family of matrices: we have that $\boldsymbol{A}_i \boldsymbol{A}_j = \boldsymbol{A}_j \boldsymbol{A}_i$, for all $i, j \in \{1, \ldots, n\}$.

Iterating the state equations leads to a multivariate generalization of the Vandermonde structure observed in the 1D case. The multivariate Vandermonde structure can again be used to determine the action matrices $\boldsymbol{A}_i$.

As described in Chapter 3, the multivariate Vandermonde shift-invariance allows us to determine the matrices $\boldsymbol{A}_i$, for $i = 1, \ldots, n$. This is a direct application of Proposition 6.1. In order to fix the ideas, let us consider an example.

Example 7.2. For example, if $n = 2$ and $d = 3$, we have that

$$\boldsymbol{M}(3)\boldsymbol{V}_{0|3}^T = \boldsymbol{0},$$

where

$$\begin{aligned}
\boldsymbol{V}_{0|3} &= \left(\begin{array}{c|cc|ccc|cccc} | & | & | & | & | & | & | & | & | & | \\ \boldsymbol{v}_{00} & \boldsymbol{v}_{10} & \boldsymbol{v}_{01} & \boldsymbol{v}_{20} & \boldsymbol{v}_{11} & \boldsymbol{v}_{02} & \boldsymbol{v}_{30} & \boldsymbol{v}_{21} & \boldsymbol{v}_{12} & \boldsymbol{v}_{03} \\ | & | & | & | & | & | & | & | & | & | \end{array}\right), \\
&= \left(\begin{array}{c|cc|c|cccc} | & | & | & & | & | & | & | \\ \boldsymbol{v}_{00} & \boldsymbol{A}_1\boldsymbol{v}_{00} & \boldsymbol{A}_2\boldsymbol{v}_{00} & \cdots & \boldsymbol{A}_1^3\boldsymbol{v}_{00} & \boldsymbol{A}_1^2\boldsymbol{A}_2\boldsymbol{v}_{00} & \boldsymbol{A}_1\boldsymbol{A}_2^2\boldsymbol{v}_{00} & \boldsymbol{A}_2^3\boldsymbol{v}_{00} \\ | & | & | & & | & | & | & | \end{array}\right).
\end{aligned}$$

From the shift-invariance property we can express $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ as follows:

$$\left(\begin{array}{c} - \boldsymbol{v}_{00} - \\ \hline - \boldsymbol{v}_{10} - \\ - \boldsymbol{v}_{01} - \\ \hline - \boldsymbol{v}_{20} - \\ - \boldsymbol{v}_{11} - \\ - \boldsymbol{v}_{02} - \end{array}\right) \boldsymbol{A}_1^T = \left(\begin{array}{c} - \boldsymbol{v}_{10} - \\ \hline - \boldsymbol{v}_{20} - \\ - \boldsymbol{v}_{11} - \\ \hline - \boldsymbol{v}_{30} - \\ - \boldsymbol{v}_{21} - \\ - \boldsymbol{v}_{12} - \end{array}\right),$$

and

$$\left(\begin{array}{c} - \boldsymbol{v}_{00} - \\ \hline - \boldsymbol{v}_{10} - \\ - \boldsymbol{v}_{01} - \\ \hline - \boldsymbol{v}_{20} - \\ - \boldsymbol{v}_{11} - \\ - \boldsymbol{v}_{02} - \end{array}\right) \boldsymbol{A}_2^T = \left(\begin{array}{c} - \boldsymbol{v}_{01} - \\ \hline - \boldsymbol{v}_{11} - \\ - \boldsymbol{v}_{02} - \\ \hline - \boldsymbol{v}_{21} - \\ - \boldsymbol{v}_{12} - \\ - \boldsymbol{v}_{03} - \end{array}\right),$$

as we have discussed in Chapter 3.

Let us illustrate this approach by means of an example.

Example 7.3. Consider the following system of polynomial equations which describes the intersection of a line and a parabola

$$\begin{array}{rclcl} f_1 & = & x_1^2 - 2x_1 - x_2 + 3 & = & 0, \\ f_2 & = & -x_1 + x_2 - 1 & = & 0. \end{array}$$

It may be checked that the two solutions are $(1,2)$ and $(2,3)$. The Macaulay matrix for this system is constructed for degree $d^\star = 3$ as

$$\boldsymbol{M}(3) = \left(\begin{array}{c|cc|ccc|cccc} 3 & -2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & -2 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & -2 & -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 \end{array}\right) \tag{7.1}$$

Let us now consider the canonical null space $\boldsymbol{H}(3)$ for this matrix, which is found as

$$\boldsymbol{H}(3) = \left(\begin{array}{cc} 1 & 0 \\ \hline 0 & 1 \\ 1 & 1 \\ \hline -2 & 3 \\ -2 & 4 \\ -1 & 5 \\ \hline -6 & 7 \\ -8 & 10 \\ -10 & 14 \\ -11 & 19 \end{array}\right). \tag{7.2}$$

By means of the shift-invariance of Example 7.2 we can extract the action matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ that will allow us to formulate the problem as a dynamical system. We find

$$\boldsymbol{A}_1^T = \begin{pmatrix} 0 & 1 \\ -2 & 3 \end{pmatrix} \text{ and } \boldsymbol{A}_2^T = \begin{pmatrix} 1 & 1 \\ -2 & 4 \end{pmatrix}.$$

From the eigenvalue decomposition of $\boldsymbol{A}_1^T$ we find $\boldsymbol{A}_1^T = \boldsymbol{V}_1\boldsymbol{D}_1\boldsymbol{V}_1^{-1}$ as follows,

$$\begin{pmatrix} 0 & 1 \\ -2 & 3 \end{pmatrix} = \begin{pmatrix} -.7071 & -.4472 \\ -.7071 & -.8944 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} -.7071 & -.4472 \\ -.7071 & -.8944 \end{pmatrix}^{-1}.$$

In the same way, we can compute $\boldsymbol{A}_2^T = \boldsymbol{V}_2\boldsymbol{D}_2\boldsymbol{V}_2^{-1}$ as

$$\begin{pmatrix} 1 & 1 \\ -2 & 4 \end{pmatrix} = \begin{pmatrix} -.7071 & -.4472 \\ -.7071 & -.8944 \end{pmatrix}\begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}\begin{pmatrix} -.7071 & -.4472 \\ -.7071 & -.8944 \end{pmatrix}^{-1}.$$

Notice that $\boldsymbol{V}_1 = \boldsymbol{V}_2$ and hence $\boldsymbol{A}_1^T\boldsymbol{A}_2^T = \boldsymbol{A}_2^T\boldsymbol{A}_1^T$.

From $\boldsymbol{H}\boldsymbol{V}_1$ and a rescaling of the columns such that the first row equals to ones, the Vandermonde structure of the null space $\boldsymbol{K}$ is reconstructed as

$$\boldsymbol{K} = \left(\begin{array}{cc} 1 & 1 \\ \hline 1 & 2 \\ 2 & 3 \\ \hline 1 & 4 \\ 2 & 6 \\ 4 & 9 \\ \hline 1 & 8 \\ 2 & 12 \\ 4 & 18 \\ 8 & 27 \end{array}\right),$$

from which the two solutions and their mutual matching can be read off as $(1, 2)$ and $(2, 3)$. The same can be done for $\boldsymbol{H}\boldsymbol{V}_2$, which yields the same result since $\boldsymbol{V}_1 = \boldsymbol{V}_2$. The initial condition $v^T(0,0)$ is read off from the first row of $\boldsymbol{H}$ as

$$v(0,0) = \begin{pmatrix} 1 & 0 \end{pmatrix}^T.$$

The complete state space description of this problem is hence

$$\begin{aligned} v(k+1,l) &= \begin{pmatrix} 0 & 1 \\ -2 & 3 \end{pmatrix} v(k,l), \\ v(k,l+1) &= \begin{pmatrix} 1 & 1 \\ -2 & 4 \end{pmatrix} v(k,l), \\ v(0,0) &= \begin{pmatrix} 1 & 0 \end{pmatrix}^T. \end{aligned}$$

Notice that when one starts from a different basis for the null space of $\boldsymbol{M}$, another system description is found. The eigenvalues of the matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ will be the same.

7.3 Solutions at Infinity: Descriptor Systems

Recall from the univariate case that roots at infinity led us to the introduction of so-called descriptor systems. In Chapter 3 we have seen that descriptor systems can elegantly be described by separating the states of the system into a regular part and a singular part, using the Kronecker canonical form (also see Appendix A). The regular part of the state has a forward-running index, whereas the singular part has a backward index.

Generalizing these concepts to the multivariate case is not straightforward. As we have seen in Chapter 5, the homogenization variable needs to be taken into account. With the homogenization variable also an additional action matrix $\boldsymbol{E}_0$ is introduced. The difficulty arises when one tries to unravel the structure of the multivariate Vandermonde state sequence matrix $\boldsymbol{W}_{0|d}$. Let us consider a very simple example to illustrate this issue.

Example 7.4. We consider the case $n = 2$ and $d = 2$. The state sequence matrix $\boldsymbol{V}_{0|2}$ is defined as

$$\boldsymbol{V}_{0|2} = \left(\begin{array}{c|cc|ccc} | & | & | & | & | & | \\ v_{00} & v_{10} & v_{01} & v_{20} & v_{11} & v_{02} \\ | & | & | & | & | & | \end{array}\right).$$

Similarly, we define the state sequence matrix $\boldsymbol{W}_{0|2}$ as

$$\boldsymbol{W}_{0|2} = \left(\begin{array}{c|cc|ccc} | & | & | & | & | & | \\ w_{200} & w_{110} & w_{101} & w_{020} & w_{011} & w_{002} \\ | & | & | & | & | & | \end{array}\right),$$

where an additional index is introduced explicitly to account for the homogenization variable x_0. Writing out the Vandermonde shift structure by means of the action matrices $\boldsymbol{E}_i$, for $i = 0, \ldots, n$, which are defined by

$$\boldsymbol{w}(k_0, \ldots, k_{j-1}, k_j - 1, k_{j+1}, \ldots, k_n) = \boldsymbol{E}_j \boldsymbol{w}(k_0, \ldots, k_n),$$

leads to the expressions

$$\begin{array}{rcccl} \boldsymbol{E}_0\boldsymbol{w}(2,0,0) & = & \boldsymbol{E}_1\boldsymbol{w}(1,1,0) & = & \boldsymbol{E}_2\boldsymbol{w}(1,0,1), \\ \boldsymbol{E}_0\boldsymbol{w}(1,1,0) & = & \boldsymbol{E}_1\boldsymbol{w}(0,2,0) & = & \boldsymbol{E}_2\boldsymbol{w}(0,1,1), \\ \boldsymbol{E}_0\boldsymbol{w}(1,0,1) & = & \boldsymbol{E}_1\boldsymbol{w}(0,1,1) & = & \boldsymbol{E}_2\boldsymbol{w}(0,0,2). \end{array}$$

An obvious consequence for the multivariate case, is that for the singular part, there is not a single 'initial condition', but rather a set of 'initial conditions', *i.e.*, the states $\boldsymbol{w}(0,2,0)$, $\boldsymbol{w}(0,1,1)$ and $\boldsymbol{w}(0,0,2)$.

In general, since the sum of the indices is always d, it is not always possible to separate the action matrices $\boldsymbol{E}_0$ and $\boldsymbol{E}_i$ with $i = 1, \ldots, n$. If the action matrices $\boldsymbol{A}_i$ and $\boldsymbol{E}_0 \backslash \boldsymbol{E}_i$, for $i = 1, \ldots, n$ can be determined, we formulate the polynomial system as a dynamical system.[1] Let us consider a few examples to fix the ideas.

Example 7.5. This example exhibits two affine roots and a double root at infinity. Consider the equations

$$\begin{array}{rcl} x_1^2 + x_1x_2 - 10 & = & 0 \\ x_2^2 + x_1x_2 - 15 & = & 0. \end{array} \qquad (7.3)$$

We construct the Macaulay matrix for degree $d = 4$ as

$$\boldsymbol{M}(4) = \begin{pmatrix} -10 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ -15 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -15 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -15 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

[1] $\boldsymbol{E}_0 \backslash \boldsymbol{E}_i$ is a simplified notation for $\boldsymbol{E}_0^{-1}\boldsymbol{E}_i$.

We compute the canonical null space as

$$\boldsymbol{H}(4)=\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 2 & 0 & 0\\ 4 & 0 & 0 & 0\\ 6 & 0 & 0 & 0\\ 9 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 10 & -1 & 0\\ 0 & 5 & 1 & 0\\ 0 & 18 & -1 & 0\\ 0 & 0 & 0 & 1\\ 40 & 0 & 0 & -1\\ 20 & 0 & 0 & 1\\ 70 & 0 & 0 & -1\\ 65 & 0 & 0 & 1 \end{array}\right),$$

from which we immediately see that the first two columns correspond to the affine roots, and the two last columns represent the roots at infinity. From the first two columns we find the action matrices $\boldsymbol{A}_1^T$ and $\boldsymbol{A}_2^T$ as in the previous examples as

$$\boldsymbol{A}_1^T=\begin{pmatrix} 0 & 1\\ 4 & 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{A}_2^T=\begin{pmatrix} 0 & 1.5\\ 6 & 0 \end{pmatrix}.$$

From the reconstruction of the null space of the affine part we find the affine roots as $(2,3)$ and $(-2,-3)$.

Again we can easily retrieve the action matrices $\boldsymbol{E}_0\backslash\boldsymbol{E}_1$ and $\boldsymbol{E}_0\backslash\boldsymbol{E}_2$ as we did for the affine part — with the difference that rows from the bottom blocks are mapped onto top rows, for example,

$$\begin{pmatrix} 0 & -1\\ 0 & 1\\ 0 & -1\\ 0 & 1\\ 1 & 0\\ -1 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0 \end{pmatrix}(\boldsymbol{E}_0\backslash\boldsymbol{E}_1)^T=\begin{pmatrix} -1 & 0\\ 1 & 0\\ -1 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0 \end{pmatrix},$$

from which we find

$$(\boldsymbol{E}_0\backslash\boldsymbol{E}_1)^T=\begin{pmatrix} 0 & 0\\ 1 & 0 \end{pmatrix}.$$

For $(\boldsymbol{E}_0\backslash\boldsymbol{E}_2)^T$, we have

$$\begin{pmatrix} 0 & 1 \\ 0 & -1 \\ 0 & 1 \\ 0 & -1 \\ -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} (\boldsymbol{E}_0\backslash\boldsymbol{E}_2)^T = \begin{pmatrix} -1 & 0 \\ 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

from which we find

$$(\boldsymbol{E}_0\backslash\boldsymbol{E}_2)^T = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}.$$

Finally, the complete state space description is

$$\begin{aligned} v(k+1,l) &= \begin{pmatrix} 0 & 1 \\ 4 & 0 \end{pmatrix} v(k,l), \\ v(k,l+1) &= \begin{pmatrix} 0 & 1.5 \\ 6 & 0 \end{pmatrix} v(k,l), \\ w(k-1,l) &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} v(k,l), \\ w(k,l-1) &= \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} v(k,l), \end{aligned}$$

with the 'initial' conditions (the ones for w have total degree 4 as the iteration of w runs backward)

$$\begin{aligned} v(0,0) &= \begin{pmatrix} 1 & 0 \end{pmatrix}^T, \\ w(4,0) &= \begin{pmatrix} 0 & 1 \end{pmatrix}^T, \\ w(3,1) &= \begin{pmatrix} 0 & -1 \end{pmatrix}^T, \\ w(2,2) &= \begin{pmatrix} 0 & 1 \end{pmatrix}^T, \\ w(1,3) &= \begin{pmatrix} 0 & -1 \end{pmatrix}^T, \\ w(0,4) &= \begin{pmatrix} 0 & 1 \end{pmatrix}^T. \end{aligned}$$

In Figure 7.1 the action matrices $\boldsymbol{A}_i$ and $\boldsymbol{E}_0\backslash\boldsymbol{E}_i$ are represented graphically, together with their initial conditions.

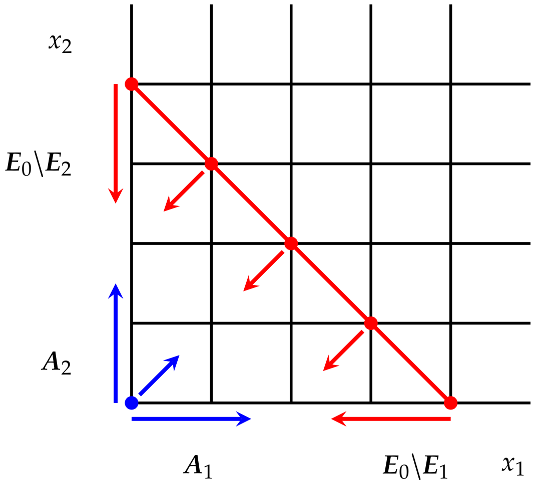

Figure 7.1: Schematic representation of the action matrices A_1, A_2, $E_0 \backslash E_1$ and $E_0 \backslash E_2$ in the monomial grid of the exponents. Multiplication with A_1 and A_2 are represented by moves to the right and upward, respectively. Multiplication with $E_0 \backslash E_1$ and $E_0 \backslash E_2$ are moves to the left and downward, respectively. The red line contains all exponents of the same degree. It is from this figure easy to understand that for the affine solutions (regular part) there is a single initial condition (indicated with the blue dot), whereas for the solutions at infinity (singular part) a set of initial conditions is required (indicated by the red dots on the red line).

Solving Over-constrained Systems 8

Systems of polynomial equations consisting of more equations than unknowns generically have no solutions. Indeed, as there are more equations than unknowns, it is unlikely that a point exists in which all equations hold. In certain situations, however, the approximate solutions of such systems are of interest, which is a relevant problem in many real-life applications.

In the current chapter the Macaulay matrix null space based SVD-based method is adapted to find the approximate solutions of an over-constrained system of polynomial equations. By investigating the smallest singular values of the Macaulay matrix, the number of approximate solutions is detected. The solutions are again obtained from the computation of the right singular vectors of the Macaulay coefficient matrix and exploiting a shift-invariance property leading to the formulation of an eigenvalue problem. The method is illustrated on several examples and an application in computer vision is discussed.

8.1 Introduction

8.1.1 Motivation

Over-constrained systems of polynomial equations consist of more equations than unknowns, and have generically no solutions. However, in many applied mathematics and engineering situations, the result of an experiment might be interpreted as a noisy realization of a set of coefficients of an underlying exact system of polynomial equations. Often it is possible to perform many of such experiments, which naturally leads to over-constrained systems of polynomial equations, of which finding the approximate solutions is of great interest.

Applications of solving over-constrained polynomial systems are found in computer vision, *e.g.*, camera pose determination (Reid et al., 2003), molecular structure determination (Emiris et al., 2006), kinematics (Bonev and Ryu, 2000) and many more.

8.1.2 Problem Formulation

We consider the problem of finding approximate solutions to an over-constrained system of polynomial equations. Consider a well-constrained system of n polynomial equations in n unknowns, formally represented as

$$\begin{array}{rcl} p_1(x_1,\ldots,x_n) & = & 0, \\ p_2(x_1,\ldots,x_n) & = & 0, \\ & \vdots & \\ p_n(x_1,\ldots,x_n) & = & 0, \end{array} \tag{8.1}$$

where $d_i := \deg(p_i)$, for $i = 1,\ldots,n$. The system (8.1) is known to have $m_B := \prod_{i=1}^{n} d_i$ affine solutions in the generic case (Cox et al., 2007).[1]

The aim of this chapter is to study the approximate solutions of an over-constrained system of $s > n$ equations in n unknowns that is a noisy realization of the system (8.1). This is formally represented as

$$\begin{array}{rcl} \rho_1(x_1,\ldots,x_n) & \approx & 0, \\ \rho_2(x_1,\ldots,x_n) & \approx & 0, \\ & \vdots & \\ \rho_s(x_1,\ldots,x_n) & \approx & 0, \end{array} \tag{8.2}$$

where $\delta_i := \deg(\rho_i)$, for $i = 1,\ldots,s$. An over-constrained system generically has no solutions in the algebraic sense, which is easy to understand as there are more equations imposing constraints than there are variables.

An instance of a case in which the search for approximate solutions to (8.2) makes sense is when we assume that each equation $\rho_i = 0$ for $i = 1,\ldots,s$ contains as its coefficients a noisy realization of the coefficients of some underlying (unknown) equation $p_i = 0$, for some $i = 1,\ldots,n$. For most of the remainder of this chapter we confine ourselves to the case that there exists such an underlying system (8.1), which is generic system and has a full support, meaning that all possible coefficients are nonzero.

8.1.3 Related Work

In the case of linear equations, the problem has well-established solution methods: the ordinary least-squares and the total least-squares problems

[1] Genericity is *e.g.*, ensured if in each equation all the possible terms occur and the coefficients are chosen randomly.

(Golub and Van Loan, 1996; Van Huffel and Vandewalle, 1991) are the numerical backbones of many state of the art parameter estimation and statistics methods. For the case of polynomial equations, however, a rather limited number of techniques to solve over-constrained systems of polynomial equations is available. The availability of such methods in off-the-shelf implementations is virtually non-existent. The discrepancy between the linear and polynomial case is an indication of the huge gap between numerical analysis and (computational) algebraic geometry, of which the latter has its origins mainly in symbolic arithmetic.

Methods in *symbolic algebra* are suitable only for finding the solutions of a well-constrained system of polynomial equations. Over-constrained systems can only be solved using symbolic methods if the existence of exact solutions is algebraically ensured (in such cases the coefficients *need* to be of infinite precision).

The currently existing methods for (approximately) solving over-constrained polynomial systems can be divided into two classes (after Ruatta et al. (2004)).

1. Methods from the first class formulate the question in a optimization framework. Dedieu and Shub (2000) employ a heuristic predictor-corrector method. Giusti and Schost (1999) reformulate the problem as the solution of a univariate polynomial. Ruatta et al. (2004) generalize the Weierstrass iteration to solve the nearest consistent system. These methods have been reported to perform well, but often need extra *a priori* knowledge, such as the number of approximate roots of interest, which is generally not available in advance.

2. The second class of methods formulates the problem using symbolic and numerical steps into eigenvalue computations of multiplication matrices operating in a basis of the quotient space. In general two methods exist to compute such multiplication matrices, namely Gröbner basis techniques and the use of Sylvester-like or Macaulay-like resultant matrices. The Gröbner basis algorithms are, however, due to their symbolic nature, not easy to modify in order to cope with over-constrained inconsistent systems.

   To the best of the authors' knowledge, only the modification of Buchberger's algorithm (Becker and Weispfenning, 1993; Buchberger, 1965) to the floating point case has been investigated in this respect, see *e.g.*, Kondratyev (2003); Shirayanagi (1996). The use of Sylvester-like and Macaulay-like resultant matrices are a natural way to deal with over-constrained inconsistent systems, as treated in Bondyfalat et al. (2000); Corless et al. (1995) (among others).

   An important step when applying the Macaulay formulation is the determination of the basis for the quotient space (*i.e.*, the normal set or standard monomials). In the classical literature this step usually proceeds by symbolic methods via the computation of a Gröbner basis or border basis, and is hence not straightforward for the case of over-constrained systems.[2]

8.2 Macaulay SVD Approach

In the current attempt we continue along the lines of the matrix-based methods and we phrase the task at hand as a numerical linear algebra problem by the use of the Macaulay coefficient matrix introduced in Chapter 5. Instead of the formulation of a Schur complement on a partitioning of a square Macaulay (sub)matrix, as in Bondyfalat et al. (2000), we will use the singular value decomposition (SVD) on the rectangular (complete) Macaulay matrix.

The use of the SVD has two big advantages over the methods discussed earlier. The classical methods require as prior knowledge at least the number of approximate solutions, and in many cases even a basis for the quotient space, in order to phrase the eigenvalue problem. The number of approximate solutions can in the Macaulay-SVD approach be obtained from counting the smallest singular values of the Macaulay matrix. Moreover, it is possible to reliably determine a basis for the set of linearly independent monomials (*i.e.,* the normal set), although this will turn out not to be necessary.

A silent assumption that is made throughout most of this chapter is that we assume that the structure of the null space of the Macaulay matrix built from the noisy over-constrained system is sufficiently similar to the multiplication structure in the null space of the Macaulay matrix of the underlying square system.

8.2.1 Detecting the Number of Approximate Solutions

An important theorem due to Weyl states that the singular values of a perturbed matrix are bounded by the perturbations.

Theorem 8.1 (Weyl (1912); Stewart (1990))**.** Let $\boldsymbol{A}$ be an $m \times n$ matrix, whose singular values are denoted $\sigma_1, \ldots, \sigma_n$. Let $\tilde{\boldsymbol{A}} := \boldsymbol{A} + \boldsymbol{E}$ denote a perturbation of the matrix $\boldsymbol{A}$, having the singular values $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n$. Then we have that

$$|\tilde{\sigma}_i - \sigma_i| \leq \|\boldsymbol{E}\|_2, \text{ for } i = 1, \ldots, n.$$

[2] The paper by Bondyfalat et al. (2000) considers the very specific case where (from *a priori* physical considerations) the number of (approximate) solutions is known to be one. In that case, it follows that the only element in the normal set is the monomial 1, and hence the step of determining the normal set by algebraic means is avoided. The case of more than one approximate solution is reported to be performed using sparse resultant theory.

This theorem can now be used to relate the separation of the singular values to the noise on the coefficients of the system. We use a similar result from Batselier et al. (2013a) to provide an upper bound of the 2-norm of the perturbation matrix $\boldsymbol{E}$ as an expression in terms of the noise on the equations. We have from Schur (1911) that

$$\|\boldsymbol{E}\|_2^2 \leq \max(r_i c_i),$$

where

$$r_i := \sum_{j=1}^{q} |e_{ij}|,$$

and

$$c_j := \sum_{i=1}^{p} |e_{ij}|.$$

For a Macaulay matrix this simplifies to (Batselier, 2013)

$$\|\boldsymbol{E}\|_2 \leq \sqrt{\sum_{i=1}^{s} \|\delta_i\|_1 \cdot \max_{1\leq i\leq s} \|\delta_i\|_1},$$

where δ_i denotes the coefficient vector corresponding to the perturbation of some equation f_i that led to ρ_i. This further simplifies to

$$\|\boldsymbol{E}\|_2 \leq \binom{n+d}{n}\epsilon\sqrt{s}, \tag{8.3}$$

where ϵ is an upper bound on the perturbation of the equations.

Under the assumption that $\binom{n+d}{n}\epsilon\sqrt{s}$ is small, this means that if the (numerically) zero singular values of the Macaulay matrix built from the well-constrained underlying system are well-separated from the nonzero singular values, this will also be the case for a perturbed system. From the following example, it can be seen that this upper bound is not always very useful in practice.

Example 8.2. Consider an over-constrained system consisting of 32 equations of degree 6 in 4 unknowns. Suppose that the maximal perturbation is $5 \cdot 10^{-4}$, then we find as an upper bound for the shift in the singular values

$$\|\boldsymbol{E}\|_2 \leq .5940.$$

This means that a singular value can shift from a numerical zero to a value of .5940, although the amount of noise on the coefficients is only of the order of magnitude of 10^{-4}.

From the experiments (reported in detail in Section 8.3), however, it turns out that the theoretical upper bound (8.3) is quite loose. We have observed

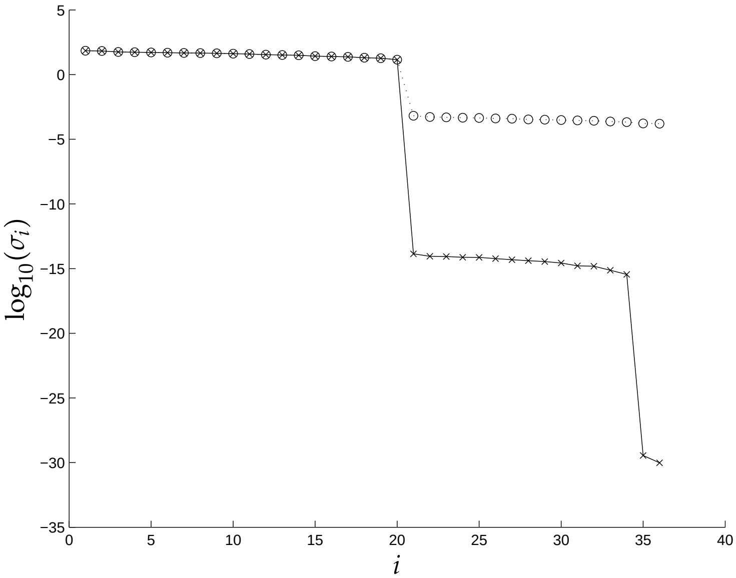

Figure 8.1: A typical plot of the spectrum of singular values of the original Macaulay matrix (full line, ×) and the Macaulay matrix built from the perturbed equations (dotted line, ∘). We observe that the singular values are strongly perturbed by the noise on the coefficients, but the separation between the (approximately) zero singular values and the nonzero singular values remains present. On the basis of the spectrum of singular values of the Macaulay matrix built from the noisy equations, the number of approximate roots can be performed.

that, for well-conditioned problems under mild noise conditions, the rank-gap is quite well maintained in the Macaulay matrix of the perturbed system and that it was possible to retrieve the number of (approximate) solutions from inspecting the singular values. For the remainder of this chapter we will assume that the separation between the singular values is indeed sufficient to determine the number of approximate solutions. In Figure 8.1 we have plotted the singular values coming from the consistent system and the singular values coming from the perturbed system for a typical example.

Inspection of the singular values of the Macaulay matrix will therefore provide the number of approximate solutions. Next, the (approximate) solutions themselves are obtained from the computation of the right singular vectors of the Macaulay coefficient matrix by exploiting the well-known shift property, leading to the formulation of an eigenvalue problem.

8.2.2 An SVD-based Approximate Solution Approach

After the number of approximate solutions m is determined, Algorithm 3 can be easily adapted to solve for the approximate solutions. First of all, an approximate null space $\boldsymbol{Z}$ is determined by collecting the right singular vectors of $\boldsymbol{M}$ corresponding to the m smallest singular values.

The determination of the standard monomials is in the over-constrained case a difficult question. However, under the genericity assumption, the standard monomials need not be determined, and one can employ the shift property on all monomial of degrees 0 up to $d^\star - 1$. This can be done because there is no influence from solutions at infinity. From the numerical experiments, it even turns out that the accuracy of the approximate roots is the best when all monomials are shifted, rather than only a subset of them.

The method for approximately solving an over-constrained generic system is summarized in Algorithm 6.

Algorithm 6. *(Approximate root-finding for over-constrained systems)*

input: system of $s > n$ 'generic' polynomials in n unknowns
$\rho_1(x_1, \ldots, x_n) \approx 0, \ldots, \rho_s(x_1, \ldots, x_n) \approx 0$
number of approximate solutions m
degree d^*

output: m approximate solutions evaluated at the function $g(\boldsymbol{x})$

1. Let $\boldsymbol{M} := \boldsymbol{M}(d^\star)$ and let $\boldsymbol{Z} := \boldsymbol{Z}(d^\star)$ denote its approximate null space (built from the right singular vectors corresponding to the m smallest singular values)

2. The m approximate roots evaluated at the shift function $g(\boldsymbol{x})$ are the eigenvalues of
$$\boldsymbol{S}_1\boldsymbol{Z}\left(\boldsymbol{T}\boldsymbol{D}_g\boldsymbol{T}^{-1}\right) = \boldsymbol{S}_g\boldsymbol{Z}$$
where $\boldsymbol{S}_1$ selects all rows of $\boldsymbol{Z}$ corresponding to the degrees 0 up to $d^\star - 1$ and $\boldsymbol{S}_g$ the mapping by $g(\boldsymbol{x})$

3. Reconstruct the mutual matching between the (approximate) solution components x_i from
$$\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{T},$$
and a consecutive column rescaling

Example 8.3. Let us consider a simple example to illustrate the method. We start from a system of two equations in two unknowns,

$$\begin{array}{rclcl} p_1(x_1, x_2) & = & x_1^3 + x_2^3 - 9x_1^2x_2 + 20x_1x_2 - 3x_1 - 20 & = & 0, \\ p_2(x_1, x_2) & = & x_1^2 + 4x_2^2 - x_1x_2 - 80 & = & 0, \end{array}$$

having six affine real solutions

x_1	x_2
−0.5942	4.3886
−0.8855	−4.5622
2.8622	−3.8943
2.9921	4.6051
−9.2118	−0.8171
9.2344	1.2729

We normalize the coefficient vectors of the system so that their 2-norm equals one. Of both the equations we now consider a noisy realization with additive noise $\sigma_n = 2 \cdot 10^{-3}$ and form an over-constrained system of four equations where of both equations two such noisy realizations are considered. We obtain the polynomials

$$\begin{aligned}
\rho_1(x_1, x_2) &= -0.671429 - 0.100961x_1 + 0.667031x_1x_2 + 0.031771x_1^3 \\
&\quad -0.299538x_1^2x_2 + 0.034994x_2^3, \\
\rho_2(x_1, x_2) &= -0.671400 - 0.100145x_1 + 0.668768x_1x_2 + 0.035137x_1^3 \\
&\quad -0.302880x_1^2x_2 + 0.036594x_2^3, \\
\rho_3(x_1, x_2) &= -1.001697 + 0.013063x_1^2 - 0.015508x_1x_2 + 0.048478x_2^2, \\
\rho_4(x_1, x_2) &= -1.000747 + 0.015580x_1^2 - 0.012807x_1x_2 + 0.050581x_2^2.
\end{aligned}$$

We then consider the Macaulay matrix of the overdetermined system and apply Algorithm 6. The approximate solutions are retrieved as

x_1	x_2
−0.6382	4.3883
−0.9352	−4.5988
2.8772	−3.8374
3.0424	4.6261
−8.6988	−0.7536
8.7327	1.2128

Although the absolute forward errors (with respect to the solutions of the underlying system) are rather large (*i.e.,* order of $.5 \cdot 10^{-1}$), we can visually verify in Figure 8.2 that the result of the procedure provides a meaningful answer to the problem.

8.3 Numerical Experiments

In a series of numerical experiments, we illustrate the performance of the method. We start from a square system having n equations in n unknowns, all

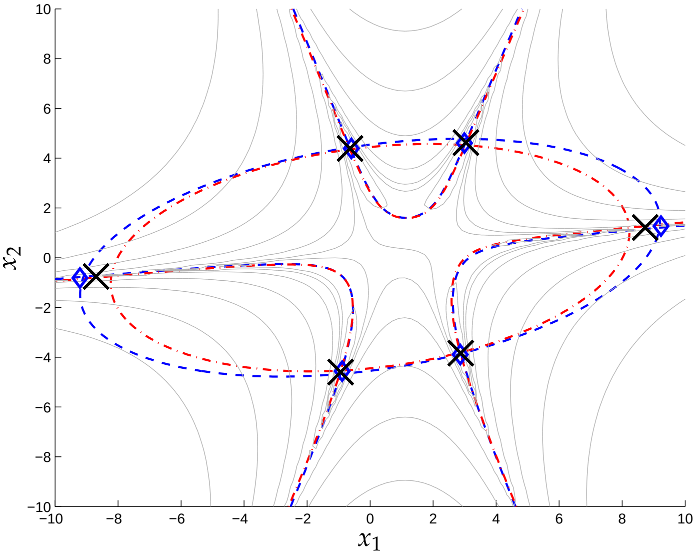

Figure 8.2: Simple example showing the result of Algorithm 6. An over-constrained system of four equations in two unknowns is shown as the dashed lines and dash-dotted lines in red and blue. The black crosses indicate the approximate solutions as retrieved by Algorithm 6. The blue diamonds represent the solutions of the underlying well-constrained system. The level sets of the least squares objective $V(x_1,\ldots,x_n) = \rho_1^2(x_1,\ldots,x_n) + \ldots + \rho_s^2(x_1,\ldots,x_n)$ are plotted in gray. It can be seen that Algorithm 6 successfully retrieves the four approximate solutions in the vicinity of the points where the curves nearly intersect and the least squares objective has its minima.

of degree $d_\circ$. The coefficients are sampled as pseudo-random integers from a discrete uniform distribution between −10 and 10, after which the coefficient vectors of all equations are normalized such that their 2-norm equals one. This system has $m_B := d_\circ^n$ solutions, which are computed with the null space based root-finding algorithm of Chapter 6.

Then an over-constrained system of $s = 3 \cdot n$ equations is constructed by introducing additive Gaussian noise (with standard deviation σ_n) to the vector of coefficients. The over-constrained system is then solved using Algorithm 6. In Table 8.1 we report

- the size of the Macaulay matrix built from the over-constrained system;
- the number of approximate solutions (*i.e.*, the Bézout number of the underlying square system);
- the average (over 5 runs) of the residual between the solution of the underlying system and the approximate solution.

For small systems, the method works well, and the residuals computed with respect to the solution of the underlying system are of the order of

Table 8.1: Numerical experiments solving over-constrained systems using Algorithm 6. We consider a number of artificially created square systems of n equations, each of degree $d_\circ$ in n variables, where the coefficients are pseudo-random integers sampled from a discrete uniform distribution between -10 and 10. Next the coefficient vectors are normalized so that their 2-norm equals 1 and we solve the square system. Then from each of the equations of the square system we build three noisy equations (using additive Gaussian noise with standard deviation σ_n) and solve the over-constrained system. We report the size of the Macaulay matrix $\boldsymbol{M}(d^\star)$ constructed from the over-constrained system (with $d^\star := n(d_\circ - 1) + 1$), the number of approximate solutions $m_B := d_\circ^n$, and the average difference between the approximate solution and the solution of the underlying system $\overline{\epsilon}$.

n	$d_\circ$	$d^\star$	σ_n	size $\boldsymbol{M}(d^\star)$	m_B	$\overline{\epsilon}$
2	2	3	$1 \cdot 10^{-5}$	18×10	4	$1.51 \cdot 10^{-6}$
2	3	5	$1 \cdot 10^{-5}$	36×21	9	$3.51 \cdot 10^{-6}$
2	4	7	$1 \cdot 10^{-5}$	60×36	16	$3.34 \cdot 10^{-6}$
2	5	9	$1 \cdot 10^{-5}$	90×55	25	$3.07 \cdot 10^{-6}$
2	6	11	$1 \cdot 10^{-5}$	126×78	36	$2.87 \cdot 10^{-6}$
2	7	13	$1 \cdot 10^{-5}$	168×105	49	$1.80 \cdot 10^{-6}$
3	2	4	$1 \cdot 10^{-5}$	90×35	8	$5.83 \cdot 10^{-5}$
3	3	7	$1 \cdot 10^{-5}$	315×120	27	$6.90 \cdot 10^{-6}$
3	4	10	$1 \cdot 10^{-5}$	756×286	64	$5.86 \cdot 10^{-5}$
3	5	13	$1 \cdot 10^{-5}$	1485×560	125	$1.82 \cdot 10^{-4}$
4	2	5	$1 \cdot 10^{-6}$	420×126	16	$1.97 \cdot 10^{-8}$
4	3	9	$1 \cdot 10^{-6}$	2520×715	81	$4.10 \cdot 10^{-7}$

the noise on the coefficients. For systems of three and four unknowns, it turned out difficult to obtain a well-conditioned problem for large degrees after introducing the perturbations. When the conditioning was ascertained, the results were satisfactory.

8.4 Application: A Computer Vision Problem

Systems of polynomial equations arise very often in a geometric context in problems in computer vision. Several of such instances are described in the PhD dissertation of Byröd (2010). We will discuss here the problem of camera pose estimation.

In the so-called problem of *camera pose estimation*, the position of a camera is to be estimated from a given set of noisy measurements, as described in Reid et al. (2003). The experiment proceeds by taking images of a reference set-up, for instance a set of n known (3D) points. Typically, for a calibration, one can perform as many such experiments as desired, which naturally leads to an over-constrained system of polynomial equations. From the over-constrained system of equations, the position of the camera can then be obtained.

Example 8.4. Let us consider the example taken from Reid et al. (2003). Suppose that we have $n = 4$ reference points A, B, C and D. A set of (noisy) images taken by the camera imposes constraints on the geometrical equations, leading to an over-constrained system. We introduce the following variables:

$$\begin{array}{rcl} p & = & 2\cos(BPC), \\ q & = & 2\cos(APC), \\ r & = & 2\cos(APB), \\ s & = & 2\cos(CPD), \\ t & = & 2\cos(APD), \\ u & = & 2\cos(BPD). \end{array}$$

We now have the over-constrained system of six equations in four unknowns

$$\begin{array}{rcl} x_1^2 + x_2^2 - rx_1x_2 - \|AB\|^2 & \approx & 0, \\ x_1^2 + x_3^2 - qx_1x_3 - \|AC\|^2 & \approx & 0, \\ x_2^2 + x_3^2 - px_2x_3 - \|BC\|^2 & \approx & 0, \\ x_1^2 + x_4^2 - sx_1x_4 - \|AD\|^2 & \approx & 0, \\ x_4^2 + x_3^2 - tx_3x_4 - \|CD\|^2 & \approx & 0, \\ x_2^2 + x_4^2 - ux_2x_4 - \|BD\|^2 & \approx & 0. \end{array}$$

Geometrically, the situation is represented in Figure 8.3.

In the example of Reid et al. (2003), the coefficients p, q, ... ,u and $\|AB\|$, ..., $\|BD\|$ are given as

$$\begin{array}{rclcrcl} p & = & -1.490710, & & \|AB\| & = & 4, \\ q & = & -.400000, & & \|AC\| & = & 8, \\ r & = & -.894427, & & \|BC\| & = & 4, \\ s & = & -1.490710, & & \|AD\| & = & 4, \\ t & = & -.666667, & & \|CD\| & = & 8, \\ u & = & -.894427, & & \|BD\| & = & 4, \end{array}$$

and are assumed to be known up to limited precision.

Applying Algorithm 6 reveals there are 4 approximate solutions to the system, appearing as two double solutions, one of which has negative entries, and the other one has positive entries:

x_1	x_2	x_3	x_4
−2.23606	−2.99999	−2.23606	−0.99362
−2.23606	−2.99999	−2.23606	−1.00636
2.23606	2.99999	2.23606	0.99362
2.23606	2.99999	2.23606	1.00636

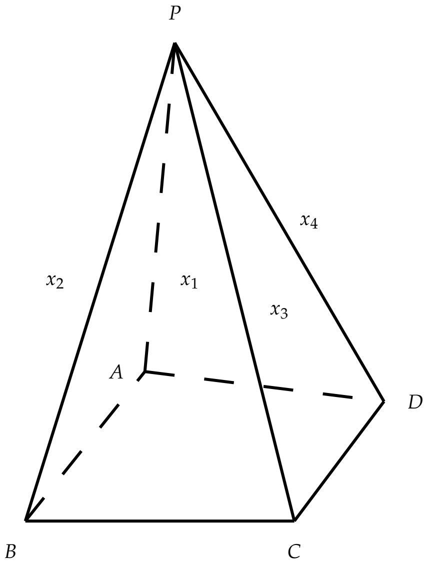

Figure 8.3: Camera pose estimation, a problem in computer vision, is concerned with determining the position of a camera on the basis of noisy measurements leading to an overdetermined system of polynomial equations. A simple instance of this problem is represented in the current diagram. A camera is centered at an unknown point P. Then a set of images is taken of four calibration points A, B, C and D, each of which leading to a constraint on the unknown camera distances $x_1, \ldots x_4$. From the geometry of this setting we then find six polynomial equations in the unknowns $x_1, \ldots, x_4$.

We were able to reduce the average of the residuals of the approximate solution obtained from averaging the two positive approximate solutions down to $1.6237 \cdot 10^{-5}$ by varying the shift function $g(\boldsymbol{x})$. On average the residual was $2.681 \cdot 10^{-5}$ for a random shift function g, measured over 50 iterations. The residual of the solution obtained from the matrix method of Reid et al. (2003) is $1.8866 \cdot 10^{-5}$.

8.5 Conclusions and Open Problems

8.5.1 Observations

The matrix-based method for solving a well-constrained system of polynomial equations turns out to be a natural starting point for solving over-constrained systems, which is a rather cumbersome task for classical computer algebra methods. Starting from a well-conditioned square system, it was shown that the matrix-based approach can successfully find approximate solutions when an over-constrained noisy realization of that system is considered.

8.5.2 Solutions at Infinity

Solutions at infinity can either be caused by the sparse support of the polynomials (*i.e.,* zero coefficients), or by the existence of algebraic relations among the coefficients of the highest degree terms. When one considers perturbations on all nonzero coefficients, solutions at infinity are in the over-constrained setting only possible when one considers equations having a sparse monomial support.

Until this point, we have dismissed the possibility of solutions at infinity, justified by the following arguments.

- The system (8.1) has $m_B := \prod_{i=1}^{n} d_i$ affine solutions, and under mild noise conditions we expect as many approximate solutions in (8.2). When the structure of (5.1), *i.e.,* d_i for $i = 1, \ldots, n$, is known, the need for a root-counting procedure is thus avoided, as well as the choice of the degree up to which the Macaulay matrix has to be built; we can take $d^\star = \sum_{i=1}^{n} d_i - n + 1$.
- Solutions at infinity caused by algebraic relations among the noisy coefficients are highly unlikely in the over-constrained case, as the algebraic relations that would have been present in the underlying equations (8.1) are most likely to be destroyed by considering a noisy realization of the coefficients.

Let us now consider how over-constrained systems with solutions at infinity can be tackled by adapting the approach developed in Chapter 6. Again, the solutions at infinity will cause standard monomials of high degree that move along to higher degrees as the degree of the Macaulay matrix is increased.

The critical part of the method, however, occurs when a suitable degree d_G needs to be chosen: unless one employs an algorithm that computes the approximate rank as we have done to determine the number of solutions, the naive approach of investigating the rank-increases in a numerical basis for the null space of the Macaulay matrix would lead to the selection of the first $m_B - m_\infty$ rows: the is a high risk that one of the subsequent approximate rank decisions is wrong, leading to a wrong choice of d_G.

However, when we assume that a suitable d_G *can* be obtained (*e.g.,* from prior knowledge of the underlying system), we can simply combine Algorithm 3 and Algorithm 6. We will illustrate this using the following example.

Example 8.5. Consider a well-constrained system of two equations in two unknowns,

$$\begin{array}{rclcl} f_1(x_1, x_2) & = & x_1 x_2 - 8 & = & 0, \\ f_2(x_1, x_2) & = & x_1^2 - 4 & = & 0, \end{array}$$

having two affine solutions $(-2,-4)$ and $(2,4)$, that can be obtained for $d_G = 4$. Since $m_B = 4$, there are also two solutions at infinity, corresponding to the standard monomials x_2^3 and x_2^4. It can be verified that it concerns a double root at infinity, described by the $(x_0,x_1,x_2) = (0,0,1)$.

Let us now consider an over-constrained system built from noisy realizations of the given equations. We consider a realization where each of the two equations is repeated two times under a perturbation of the coefficients by $\delta \sim \mathcal{N}(0, 1 \cdot 10^{-4})$ and obtain the following four equations:

$$\begin{array}{rcll}
\rho_1(x_1,x_2) & = & 0.99996x_1x_2 - 7.99998 & \approx & 0, \\
\rho_2(x_1,x_2) & = & 1.00014x_1x_2 - 7.99997 & \approx & 0, \\
\rho_3(x_1,x_2) & = & 1.00015x_1^2 - 3.99999 & \approx & 0, \\
\rho_4(x_1,x_2) & = & 1.00005x_1^2 - 3.99993 & \approx & 0.
\end{array}$$

We construct the Macaulay matrix for the over-constrained system for degree $d_G = 4$ and denote it by $\boldsymbol{M}$. Again the standard monomials x_2^3 and x_2^4 appear. Now a column compression of the numerical basis for the null space of $\boldsymbol{M}$ is computed, which we denote by $\boldsymbol{W}$. After retaining from $\boldsymbol{W}$ the part that corresponds to the affine solutions (as in Chapter 6), we write the shift relation (Proposition 6.3) for a random shift polynomial $g(x_1,x_2)$, leading to the approximate solutions $(1.99989, 4.00001)$ and $(-1.99989, -4.00001)$.

8.5.3 Recovering Underlying System

We have limited the pursuit of solving overdetermined systems to the case where we find a (dense) low-rank approximation of the Macaulay matrix, denoted by $\widetilde{\boldsymbol{M}}(d)$, which has an exact null space, and we assume that the null space of $\widetilde{\boldsymbol{M}}(d)$ has a structure sufficiently close to the multiplication structure needed to phrase the eigenvalue problem.

By discarding the smallest singular values of the Macaulay matrix $\boldsymbol{M}(d)$, the interpretation which system we are really solving is lost. In specific cases, one may indeed be interested in recovering some the underlying 'exact' well-constrained system of equations of which the over-constrained system is a noisy realization. The correct way to go about in this situation is to find a structured low-rank approximation of the noisy Macaulay matrix, from which a set of s (or n) equations can be retrieved, having *exact* solutions.

This question reduces to solving a structured total least squares problem, which is discussed in Appendix A, where the Macaulay structure should be imposed in the constraints. Let $\boldsymbol{M} := \boldsymbol{M}(d^\star)$ denote the $p \times q$ Macaulay matrix built from the over-constrained system of equations. Let $\boldsymbol{M}_{\boldsymbol{\theta}}$ represent a $p \times q$ parametrized Macaulay matrix where the parameter vector $\boldsymbol{\theta}$ contains the coefficients of the equations that need to be recovered. The search for the coefficients can be formulated as the minimization problem

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \boldsymbol{\theta}^T\boldsymbol{\theta}, \\ \text{subject to} \quad & \operatorname{rank}(\boldsymbol{M}_{\boldsymbol{\theta}}) = q - m_B. \end{aligned}$$

8.5.4 Conditioning of the System

An important factor influencing the 'good-ness' of a solution is given by the conditioning of the *problem*. In this respect there are in fact two conditioning aspects we need to consider.[3]

- First of all, the conditioning of the polynomial evaluation is of importance, as a small variation of a computed root may give a dramatic change in the evaluation in one of the equations ρ_i. We have in general

$$K_e := \lim_{\epsilon \to 0+} \sup_{\|\delta \boldsymbol{x}\| \leq \epsilon} \frac{\|\boldsymbol{\rho}(\boldsymbol{x}+\delta\boldsymbol{x}) - \boldsymbol{\rho}(\boldsymbol{x})\|}{\|\boldsymbol{\rho}(\boldsymbol{x})\|} \Bigg/ \frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|},$$

  where $\boldsymbol{\rho}(\cdot)$ represents the vectorization of the polynomials $\rho_i, i = 1, \dots, s$. This leads to

$$K_e = \frac{\|J\|}{\|\boldsymbol{\rho}(\boldsymbol{x})\|/\|\boldsymbol{x}\|},$$

  where J denotes the Jacobian matrix (1.2) containing the partial derivatives of the equations $\rho_i, i = 1, \dots, s$ with respect to the unknowns $x_1, \dots, x_n$.

- Secondly, the conditioning of the polynomial system solving problem itself is of importance. It can be shown (Stetter, 2004) that the amount of change in the solutions of a problem are related to the change in the coefficients. Let $F(\boldsymbol{a})$ represent an implicit expression a certain root depending on the (noisy) coefficients $\boldsymbol{a}$. The evaluation of an equation ρ_i is then given by $\rho_i(F(\boldsymbol{a}; a)$, where the explicit dependence of the coefficients is emphasized. By carefully working out the chain rule for differentiation, an expression of the condition number is obtained as

$$K_s = \left\| \frac{\partial \rho}{\partial x} \left(F(\boldsymbol{a}); \boldsymbol{a}\right)^{-1} \right\| \cdot \left\| \frac{\partial \rho}{\boldsymbol{a}} \left(F(\boldsymbol{a}); \boldsymbol{a}\right) \right\|.$$

  The factor $\| \frac{\partial \rho}{\partial x} \left(F(\boldsymbol{a}); \boldsymbol{a}\right)^{-1} \|$ is the norm of the inverse of the Jacobian matrix (see (1.2)) evaluated at the root under consideration.

[3] It must be emphasized that in what follows, the influence of the conditioning makes abstraction of the question of correctly determining the *number of solutions* by investigating the singular values. Correctly estimating the number of approximate solutions is of paramount importance.

In the numerical experiments, the inspection of the Jacobian matrix and its singular values was used to detect poor numerical conditions. A detailed error analysis has not yet been performed, but the methods proposed in Stetter (2004) seem to be a good starting point to further investigate this matter.

Part III

Closing

Conclusions and Outlook 9

9.1 Conclusions

Algebraic geometry is perhaps the most fundamental branch of mathematics, with a rich history and results that pervade all fields of mathematics and engineering. Although it has turned into a field of mainly theoretical research for about a century, the last decades have witnessed a renewed interest in its computational and algorithmic results thanks to the ever increasing advances in computer science. Due to its theoretical nature, many algorithms are not directly suited for the implementation on computing devices, witnessed by the cumbersome generalization of Gröbner basis computations to floating point environments. This manuscript makes a humble 'first' effort in bridging the worlds of abstract polynomial algebra and numerical computation.

We have presented a framework to tackle polynomial system solving from the linear algebra point of view. The problem at hand is phrased as a question of linear algebra by the definition of the Macaulay matrix. The polynomial system solving problem can then be rephrased as an eigenvalue problem by making use of the monomial structure present in the null space of the Macaulay matrix. This gives rise to the first so-called null space based solution method. Another way to approach the problem is to investigate the linearly dependent and independent columns of the Macaulay matrix. It turns out possible to phrase the polynomial system solving problem as a problem in the columns of the Macaulay matrix only. This leads to a (Q-less) QR decomposition on a column-reordered version of the Macaulay matrix, from which the solutions of the system follow from eigendecompositions.

The observation that there is an intimate link between polynomial system solving and eigenvalue problems is not new at all, and important numerical aspects are elaborately worked out in great detail in the work by Stetter (2004). This was a huge source of inspiration for our research, but it assumes that a set of standard monomials (*i.e.,* normal set) is available. Stetter claims that a Gröbner basis is one of the ways to determine a normal set, and at the time the *only* way to effectively compute it using widely available software implementations. The work on border bases (Mourrain, 1999; Kehrein and Kreuzer, 2006) will supposedly provide a numerically more sound approach, however this pursuit is at the moment mostly of academic nature, and the 'algorithms' that are being developed are theoretical.

This manuscript differs with the classical work on the link between polynomial system solving and eigenvalue problems in exactly this: in this text the notion of the standard monomials (the normal set) naturally follows from the problem, but it does not play a central role in the algorithms. We have shown that all choices regarding the number of solutions, certain selection of rows, *etc.*, can be incorporated in numerically reliable rank decisions. During the construction of the eigenvalue problem for finding the solutions of a system, by selecting all degree blocks that contain standard monomials, the risk of making a *wrong choice* of standard monomials is avoided.

Moreover, we have chosen to employ a specific monomial ordering throughout the text, however, from the numerical linear algebra point of view, any graded ordering would yield exactly the same results, as this merely represents a column-permutation of the Macaulay matrix, and hence our methods are not sensitive to a change of ordering as it is the case for Gröbner basis algorithms.

Another important observation is the intimate links between polynomial system solving and multidimensional realization theory, as pointed out first by Hanzon and Hazewinkel (2006). Our method differs in the sense that the matrix-based approach presented in this text does not require the construction of a Gröbner basis to phrase the polynomial system as a state space model consisting of multivariate difference equations. Indeed, it turns out that the Macaulay matrix and its null space serve as the interfaces between a given polynomial system and its interpretation as a multivariate system. It is expected that the conceptual links between polynomial system solving and multidimensional systems will turn out fruitful. For instance, we foresee that certain fundamental algebraic geometry results will have their system theoretic counterparts, such as the Hilbert regularity of a system and a variation of the Cayley-Hamilton theorem operating on the null space of the Macaulay matrix.

The numerical linear algebra approach allowed us to tackle the problem of over-constrained polynomial systems, which is a rather cumbersome task to tackle using the classical computer algebra methods. The task of finding approximate solutions to an over-constrained system of polynomials occurs often in applied mathematics and engineering contexts where noisy measurements of a phenomenon lead to an overdetermined system of equations. We showed that a naive way of phrasing the task led to competitive results and pointed out some ideas for future research.

We have worked out a number toy problems of possible application fields to show the potential application fields of our work. The first area is the application of the theory to polynomial optimization problems, where we discussed applications in the area of system identification. We have illustrated that the structured total least squares problem, which is an important tool in system identification, boils down to solving a polynomial optimization problem, or solving a system of polynomial equations. It was shown that the approach presented in the thesis is able to solve small, yet nontrivial instances of this problem. For simple models involving a small number of data points, our approach is able to retrieve the globally optimal model. The method for solving over-constrained polynomial systems has been applied to a problem from computer vision, namely camera pose determination on the basis of a set of noisy measurements. The approach developed in Chapter 8 turned out to be competitive with the methods reported in the literature.

9.2 Future Work

In general, although the body of literature in (computational) algebraic geometry is vast, we strongly feel that the bridging between polynomial algebra and numerical analysis has only merely begun. Indeed, after several decades of developing linear modeling into a *technology*, the natural next step is turning the attention to polynomial models.

George Bernard Shaw once said that "Science never solves a problem without creating ten more." This must be one of the most exciting aspects of doing research, and it is ever so true in a research field that has been explored only recently.

The work in this thesis has provided a new framework for tackling certain questions in polynomial algebra — and at the same time, a multitude of 'new' challenges have been coined and remain unanswered. In the following paragraphs a personal view is outlined along which lines the future work should be organized.

9.2.1 Solutions at Infinity and Multiplicities in Realization Theory Framework

An important step in the future research is to work out the relation of the multiplicity of a solution to the multiple eigenvalue problem as in Dayton et al. (2011); Marinari et al. (1996); Möller and Stetter (1995) and to work it into the realization theory framework. Although they are often not of practical interest, some work should be done on phrasing the solutions at infinity in the realization theory based framework. The first step, namely developing an understanding of the case of multiple solutions, is essential, as in most cases, solutions at infinity occur with multiplicity.

The realization theory framework allows natural conceptual interpretations of multivariate polynomial systems. An exciting question is to investigate whether the notion of the (Hilbert) regularity of an ideal (*i.e.,* the degree at which the Hilbert function becomes polynomial (Cox et al., 2007)) has an equivalent system theoretical interpretation: does this have a relation to the degrees at which the Macaulay matrix becomes sufficiently informative enough to obtain the solutions from the nilpotency of the part of the singular parts? Such a notion would correspond closely to a variation of the Cayley-Hamilton theorem.

9.2.2 Developing Understanding of Numerical Aspects

Whereas the SVD-based Macaulay matrix approach from the numerical point of view is rather safe, the repercussions on the numerics are rather poorly understood. As such, the current thesis may be seen as a starting point, where several interesting connections are established. However, from the numerical point of view, this is merely a first step; much work can be expected to phrase the ideas developed in this thesis into truly efficient and reliable methods.

It is of paramount importance to develop an understanding of the numerical aspects of the presented framework. The work by Stetter (2004) and Jónsson and Vavasis (2004) may serve as starting points for this pursuit, as they have taken important steps in understanding the numerics behind the eigenvalue-approach to the polynomial system solving problem. However, in both approaches, the problem is formulated in a different manner as in our case.

9.2.3 Over-constrained Systems

The work on over-constrained systems is a whole research avenue on its own with a multitude of potential applications in engineering and applied mathematics problems. As we already mentioned, there is an enormous gap between the case of linear overdetermined systems (with well-known solutions such as (total) least squares approaches) and polynomial overdetermined systems. As we have shown in Chapter 8 the presented framework seems a natural starting point for this problem. There are still many challenges and unanswered questions in this problem, such as deriving a theoretical (or empirical) upper bound for the required degree of the Macaulay matrix, understanding numerical aspects and conditioning, *etc.*

9.2.4 Identifiability Analysis of Nonlinear Systems

In the work of Ljung and Glad (1994) an algorithm based on differential algebra was presented to analyze the identifiability of model parameters in certain nonlinear dynamical models. Models that can be analyzed in this framework are polynomial differential equations in which the data and the model parameters may occur as (differentials of) polynomial functions. The analysis makes use of Ritt's algorithm. Loosely speaking, Ritt's algorithm is the differential algebra variation of Buchberger's polynomial algebra algorithm. The analysis of Ljung and Glad (1994) was done for continuous-time models, and in recent work by Lyzell et al. (2011) a similar method was tailored towards discrete-time systems. It would be interesting to investigate how the methods developed in this manuscript may become of use in this application field.

9.2.5 Numerical Basis Computation

An important link that has not been investigated in this thesis is the connection between the Macaulay matrix approach and the so-called border bases approaches (Mourrain, 1999; Kehrein and Kreuzer, 2006). As opposed to Gröbner bases, border bases have superior numerical properties. For example, border bases can be constructed such that they continuously depend on the coefficients of the system; introducing a small nonzero coefficient does not dramatically change the border basis, whereas the introduction of a tiny nonzero coefficient may lead to a completely different (and often, numerically ill-posed) Gröbner basis. Our approach in not explicitly requiring a set of standard monomials does seem to correspond closely to the basic ideas behind border bases.

9.2.6 Sparsity, Structure and Real Solutions

It must be emphasized that nearly all methods in polynomial system solving and polynomial optimization suffer from the well-known explosive increase of the number of monomials as more variables (or higher degrees) are considered.

There are two important approaches from the literature that might prove relevant:

1. First of all, although the monomial expansion is an inherent property of the task at hand, it is of great interest to alleviate these effects by considering the relation to sparse resultants (Emiris and Mourrain, 1999b), and devising ways to exploit the structure and sparsity in the matrix computations.

2. Secondly, recent advances in real algebraic geometry (Lasserre et al., 2012; Laurent and Rostalski, 2012; Parrilo, 2000) have led to very competitive methods for solving polynomial optimization problems, such as the well-known sums-of-squares polynomials.

Moreover, the case of real root-finding and directly solving for the minimizing root of a certain objective is of particular interest in polynomial optimization; the work by Hanzon and Jibetean (2003); Lasserre et al. (2012); Laurent and Rostalski (2012); Parrilo (2000) is likely to be relevant in this context.

The ultimate goal for the problem of solving a polynomial optimization problem is a method in which the null space of the Macaulay matrix is never explicitly calculated. By using the most economic representation of a system, *i.e.,* the coefficients of the system, intelligent FFT-like computations operating on the coefficients that implicitly employ the quasi-Toeplitz block structure of the Macaulay matrix could ultimately lead to a direct power method-like solution of the eigenvalue computation, returning of the (real) root of interest only.

Bibliography

R. J. Adcock. Note on the method of Least Squares. *The Analyst*, IV(6):183–184, Nov. 1877.

R. J. Adcock. A problem in Least Squares. *The Analyst*, V(2):53–54, Mar. 1878.

S. Attasi. Modelling and recursive estimation for double indexed sequences. System Identification: Advances and Case Studies, pages 289–348. Academic Press, New York, 1976.

W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. Proc. Int. Conf. Num. Math., pages 11–30. Birkhäuser, 1988.

W. Auzinger and H. J. Stetter. A study of numerical elimination for the solution of multivariate polynomial systems. Technical report, Institut für Angewandte und Numerische Mathematik, TU Wien, 1989.

K. Batselier. *A Numerical Linear Algebra Framework for Solving Problems with Multivariate Polynomials*. PhD thesis, KU Leuven, 2013.

K. Batselier, P. Dreesen, and B. De Moor. A geometrical approach to finding multivariate approximate LCMs and GCDs. *Lin. Alg. Appl.*, 438(9):3618–3628, May 2013a.

K. Batselier, P. Dreesen, and B. De Moor. The geometry of multivariate polynomial division and elimination. *SIAM J. Mat. Anal. Appl.*, 34(1):102–125, 2013b.

T. Becker and V. Weispfenning. *Gröbner Bases: A Computational Approach to Commutative Algebra*. Springer Verlag, New York, 1993.

I. Bleylevens, R. Peeters, and B. Hanzon. Efficiency improvement in an nD-systems approach to polynomial optimization. *J. Symb. Comput.*, 42:30–53, 2007.

D. Bondyfalat, B. Mourrain, and V. Pan. Solution of a polynomial system of equations via the eigenvector computation. *Lin. Alg. Appl.*, pages 193–209, 2000.

I. A. Bonev and J. Ryu. A new method for solving the direct kinematics of general 6-6 stewart platforms using three linear extra sensors. *Mech. Mach. Theor.*, 35(3):423–436, 2000.

B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.

B. Buchberger. Gröbner bases and systems theory. *Multidimens. Syst. Signal Process.*, 12:223–251, 2001.

M. Byröd. *Numerical Methods for Geometric Vision: From Minimal to Large Scale Problems*. PhD thesis, Lund University, Faculty of Engineering, Centre for Mathematical Sciences, Lund, Sweden, 2010.

R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. Proc. 1995 Int. Symp. Symb. Algebraic Comput. (ISSAC), pages 195–207. ACM Press, 1995.

D. A. Cox, J. B. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer-Verlag, New York, second edition, 2005.

D. A. Cox, J. B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Springer-Verlag, third edition, 2007.

B. H. Dayton, T.-Y. Li, and Z. Zeng. Multiple zeros of nonlinear systems. *Math. Comp.*, 80:2143–2168, 2011.

B. De Moor. Structured Total Least Squares and L_2 approximation problems. *Lin. Alg. Appl.*, 188–189:163–207, 1993.

B. De Moor. Dynamic Total Linear Least Squares. *Proc. 10th IFAC Symp. Syst. Identif.*, 3:159–164, 1994a.

B. De Moor. Total Least Squares for affinely structured matrices and the noisy realization problem. *IEEE Trans. Signal Process.*, 42(11):3104–3113, 1994b.

J. P. Dedieu and M. Shub. Newton's method for overdetermined systems of equations. *Math. Comp.*, 69(231):1099–1115, 2000.

A. Dickenstein and I.Z. Emiris, editors. *Solving Polynomial Equations*, volume 14 of *Algorithms and Computation in Mathematics*. Springer, 2005.

P. Dreesen and B. De Moor. Polynomial optimization problems are eigenvalue problems. In P. M. J. Van den Hof, C. Scherer, and P. S. C. Heuberger, editors, *Model-Based Control – Bridging Rigorous Theory and Advanced Technology*, pages 49–68. Springer, 2009.

P. Dreesen, K. Batselier, and B. De Moor. Back to the roots: polynomial system solving, linear algebra, systems theory. Proc. 16th IFAC Symp. Syst. Ident. (SYSID 2012), pages 1203–1208, 2012a.

P. Dreesen, K. Batselier, and B. De Moor. Back to the roots – polynomial system solving using linear algebra. Technical Report 12-169, KU Leuven Department of Electrical Engineering ESAT/SCD, 2012b.

P. Dreesen, K. Batselier, and B. De Moor. Weighted/structured total least squares problems and polynomial system solving. 11th Eur. Symp. Artif. Neur. Netw., Comput. Intell. Mach. Learn. (ESANN 2012), pages 351–356, 2012c.

P. Dreesen, K. Batselier, and B. De Moor. Solving overconstrained polynomial systems. Technical Report 13-103, KU Leuven Department of Electrical Engineering ESAT/SCD, 2013a.

P. Dreesen, K. Batselier, and B. De Moor. Polynomial system solving and nD realization theory. Technical Report 12-170, KU Leuven Department of Electrical Engineering ESAT/SCD, 2013b.

G. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychom.*, 1:211–218, 1936.

I. Z. Emiris. *Sparse Elimination and Applications in Kinematics*. PhD thesis, UC Berkeley, Dec 1994.

I. Z. Emiris and B. Mourrain. Computer algebra methods for studying and computing molecular conformations. *Algorithm.*, 25:372–402, 1999a.

I. Z. Emiris and B. Mourrain. Matrices in elimination theory. *J. Symb. Comput.*, 28(1–2):3–44, 1999b.

I. Z. Emiris, E. D. Fritzilas, and D. Manocha. Algebraic algorithms for structure determination in biological chemistry. *Int. J. Quantum Chem.*, 106(1):190–210, 2006.

J. C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra*, 139(1):61–88, 1999.

J. C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Proc. 2002 Int. Symp. Symb. Algebraic Comput. (ISSAC), pages 75–83. ACM Press, 2002.

K. Gałkowski. *State-space Realizations of Linear 2-D Systems with Extensions to the General nD ($n > 2$) case*. Lecture Notes in Control and Information Sciences. Springer, 2001.

F. Gantmacher. *The Theory of Matrices, volume* 2. Chelsea Publishing Company, New York, 1960.

C. F. Gauss. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*. F. Perthes and I. H. Besser, Hamburg, 1809.

I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Birkhäuser, Boston, 1994.

M. Gerdin. Computation of a canonical form for linear differential-algebraic equations. In *Proc. of Regl.*, Göteborg, May 2004a.

M. Gerdin. *Identification and Estimation for Models Described by Differential-Algebraic Equations*. PhD thesis, Linköping University, Department of Electrical Engineering, Division of Automatic Control, Linköping, Sweden, 2004b.

M. Giusti and E. Schost. Solving some overdetermined polynomial systems. Proc. 1999 Int. Symp. Symb. Algebraic Comput. (ISSAC), pages 1–8. ACM, 1999.

G. H. Golub and C. F. Van Loan. An analysis of the Total Least Squares problem. *SIAM J. Numer. Anal.*, 15(17):883–893, 1980.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

B. Hanzon and M. Hazewinkel. An introduction to constructive algebra and systems theory. In B. Hanzon and M. Hazewinkel, editors, *Constructive Algebra and Systems Theory*, pages 2–7. Royal Netherlands Academy of Arts and Sciences, 2006.

B. Hanzon and D. Jibetean. Global minimization of a multivariate polynomial using matrix methods. *J. Glob. Optim.*, 27:1–23, 2003.

D. Hilbert. Über die Theorie der algebraischen Formen. *Math. Ann.*, 36:473–534, 1890.

B. L. Ho and R. E. Kalman. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 14(12):545–548, 1966.

A. S. Householder and G. Young. Matrix approximation and latent roots. *Americ. Math. Mon.*, pages 165–171, 1938.

G. F. Jónsson and S. A. Vavasis. Accurate solution of polynomial equations using Macaulay resultant matrices. *Math. Comput.*, 74(249):221–262, 2004.

T. Kailath. *Linear Systems*. Prentice-Hall information and system sciences series. Prentice-Hall International, 1998.

A. Kehrein and M. Kreuzer. Computing border bases. *J. Pure Appl. Alg.*, 205: 279–295, 2006.

A. Kondratyev. *Numerical Computation of Gröbner Bases*. PhD thesis, Johannes Kepler Universität Linz, 2003.

L. Kronecker. Algebraische Reduction der Schaaren bilinearer Formen. *Monatshefte Akad. Wissensch Berlin*, pages 1225–1237, 1890. Reprinted in Leopold Kronecker's Werke, Chelsea, 1968, pp. 139–155.

S. Y. Kung. A new identification and model reduction algorithm via Singular Value Decomposition. Proc. 12th Asilomar Conf. Circuits, Syst. Comput., pages 705–714, Pacific Grove, CA, 1978.

J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001.

J. B. Lasserre, M. Laurent, B. Mourrain, P. Rostalski, and P. Trébuchet. Moment matrices, border basis and real radical computation. *J. Symb. Comput.*, 2012.

M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research & Management Science*. Springer-Verlag, 2012.

D. Lazard. Résolution des systèmes d'équations algébriques. *Theor. Comput. Sci.*, 15:77–110, 1981.

D. Lazard. Groebner bases, Gaussian elimination and resolution of systems of algebraic equations. In J. van Hulzen, editor, *Computer Algebra*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin / Heidelberg, 1983.

A.-M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. Courcier, Paris, 1806.

P. Lemmerling. *Structured Total Least Squares: Analysis, algorithms and applications*. PhD thesis, Katholieke Universiteit Leuven, Faculteit Toegepaste Wetenschappen, Departement Elektrotechniek, Kasteelpark Arenberg 10, 3001 Leuven (Heverlee), May 1999.

P. Lemmerling and B. De Moor. Misfit versus Latency. *Autom.*, 37:2057–2067, 2001.

T. Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numer.*, 6:399–436, 1997.

L. Ljung. *System identification: theory for the user*. Prentice Hall PTR, Upper Saddle River, NJ, 1999.

L. Ljung and T. Glad. On global identifiability for arbitrary model parametrizations. *Autom.*, 35(2):265–276, 1994.

C. Lyzell, T. Glad, M. Enqvist, and L. Ljung. Difference algebra and system identification. *Autom.*, 47(9):1896–1904, 2011.

F. S. Macaulay. On some formulae in elimination. *Proc. London Math. Soc.*, 35: 3–27, 1902.

F. S. Macaulay. *The algebraic theory of modular systems*. Cambridge University Press, 1916.

D. Manocha. Solving systems of polynomial equations. *IEEE Comput. Graph. Appl.*, 14(2):46–55, 1994.

M. G. Marinari, H. M. Möller, and T. Mora. On multiplicities in polynomial system solving. *Trans. Amer. Math. Soc.*, 348:3283–3321, 1996.

I. Markovsky. Structured low-rank approximation and its applications. *Autom.*, 44:891–909, 2008.

H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numer. Math.*, 70:311–329, 1995.

M. Moonen, B. De Moor, J. Ramos, and S. Tan. A subspace identification algorithm for descriptor systems. *Syst. Contr. Lett.*, 19:47–52, 1992.

T. Mora. *Solving Polynomial Equation Systems I: The Kronecker-Duval Philosophy*, volume 88 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003.

T. Mora. *Solving Polynomial Equation Systems II: Macaulay's Paradigm and Gröbner Technology*, volume 88 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2005.

T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. The double description method. In *Contributions to the theory of games, vol. 2*, Annals of Mathematics Studies, no. 28, pages 51–73. Princeton University Press, Princeton, N. J., 1953.

B. Mourrain. A new criterion for normal form algorithms. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 1710 of *Lecture Notes in Computer Science*, pages 430–443. Springer, Berlin, 1999.

B. Mourrain and V. Y. Pan. Multivariate polynomials, duality, and structured matrices. *J. Complex.*, 16(1):110–180, 2000.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

L. Pachter and B. Sturmfels. *Algebraic Statistics for Computational Biology*. Cambridge, 2005.

V. Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Rev.*, 39(2):187–220, June 1997.

P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 2000.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Phil. Mag.*, 2(6), 1901.

R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach*. Wiley-IEEE Press, 2nd edition, 2012.

G. Pistone, E. Riccomango, and H. P. Wynn. *Algebraic Statistics*. CRC Press, 2001.

G. Reid, J. Tang, and L. Zhi. A complete symbolic-numeric linear method for camera pose determination. Proc. 1997 Int. Symp. Symb. Algebraic Comput. (ISSAC), pages 215–223, New York, NY, USA, 2003. ACM.

J. B. Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM J. Matrix Anal. Appl.*, 17:110–126, 1996.

O. Ruatta, M. Sciabica, and A. Szanto. Over-constrained Weierstrass iteration and the nearest consistent system. Technical Report RR-5215, INRIA, 2004. Accepted to the Journal of Complexity.

T. Sasaki and F. Kako. Term cancellations in computing floating-point Gröbner bases. In V. P. Gerdt, W. Koepf, E. W. Mayr, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 6244 of *Lecture Notes in Computer Science*, pages 220–231, Tsakhkadzor, Armenia, Sep 2010. 12th CASC International Workshop.

I. Schur. Bemerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen. *J. Reine Angew. Math.*, 140:1–28, 1911.

E. Serpedin and G. B. Giannakis. A simple proof of a known blind channel identifiability result. *IEEE Trans. Signal Process.*, 47:591–593, 1999.

K. Shirayanagi. An algorithm to compute floating-point Gröbner bases. In T. Lee, editor, *Mathematical Computation with Maple V: Ideas and Applications*, pages 95–106, Univ Michigan, Ann Arbor, MI, 1993.

K. Shirayanagi. Floating point Gröbner bases. *J. Math. Comput. Simul.*, 42(4-6): 509–528, Jan 1996.

N. Z. Shor. Class of global minimum bounds of polynomial functions. *Cybern.*, 23(6):731–734, 1987.

N. Z. Shor and P. I. Stetsyuk. The use of a modification of the r-algorithm for finding the global minimum of polynomial functions. *Cybern. Syst. Anal.*, 33:482–497, 1997.

D. E. Smith. *History of Mathematics, Volume 1*. Dover, New York, 1951.

D. E. Smith. *History of Mathematics, Volume 2*. Dover, New York, 1953.

T. Söderström and P. Stoica. *System Identification*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

H. J. Stetter. Matrix eigenproblems are at the heart of polynomial system solving. *ACM SIGSAM Bull.*, 30(4):22–25, 1996.

H. J. Stetter. Stabilization of polynomial systems solving with Groebner bases. Proc. 1997 Int. Symp. Symb. Algebraic Comput. (ISSAC), pages 117–124, New York, NY, USA, 1997. ACM.

H. J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.

G. W. Stewart. Perturbation Theory for the Singular Value Decomposition. In R. J. Vaccaro, editor, *SVD and Signal Processing, II: Algorithms, Analysis and Applications*, pages 99–109. Elsevier Science, 1990.

G. W. Stewart. On the early history of the singular value decomposition. *SIAM Rev.*, 35:551–566, 1993.

G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 4th edition edition, 2009.

B. Sturmfels. Solving systems of polynomial equations. Number 97 in CBMS Regional Conference Series in Mathematics, Providence, 2002. American Mathematical Society.

J. J. Sylvester. On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's function and that of the greatest algebraical common measure. *Trans. Roy. Soc. Lond.*, 1853.

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

B. van der Waerden. *Moderne Algebra, Volume II*. Springer Verlag, Berlin, 1931.

S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1991.

P. Van Overschee and B. De Moor. *Subspace identification for linear systems: theory, implementation, applications*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.

J. Verschelde. *Homotopy Continuation Methods for Solving Polynomial Systems*. PhD thesis, KU Leuven, 1996.

J. Verschelde. Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999.

K. Weierstrass. Zur Theorie der bilinearen und quadratische Formen. *Monatshefte Akademie der Wissenschaften Berlin*, pages 310–338, 1868.

Reprinted in Mathematische Werke von Karl Weierstrass 2, pp. 19–44, Mayer & Mueller, Berlin, 1895.

H. Weyl. Das asymptotische Verteilungsgestez der Eigenwert linearer partieller Differentialgleichungen (mit einer Anwendung auf der Theorie der Hohlraumstrahlung). *Math. Ann.*, 71:441–479, 1912.

J. C. Willems. From time series to linear system – Part I. Finite dimensional linear time invariant systems. *Autom.*, 22(5):561–580, 1986a.

J. C. Willems. From time series to linear system – Part II. Exact modeling. *Autom.*, 22(6):675–694, 1986b.

J. C. Willems. From time series to linear system – Part III. Approximate modeling. *Autom.*, 23(1):87–115, 1987.

Part IV

Appendices

Linear Algebra and Systems Theory

A

In the current chapter the fundamental tools of (numerical) linear algebra are reviewed, such as vectors and matrices, the rank of a matrix, the fundamental subspaces of a matrix, the QR, SVD and eigendecomposition of a matrix. Many of the notions here are based on the classical text books Golub and Van Loan (1996); Strang (2009); Trefethen and Bau (1997). These works are excellent resources providing a complete treatment of matrix computations and numerical linear algebra.

Furthermore, we will briefly review some essential concepts coming from realization theory. The discussion will start with the essentials of dynamical linear time-invariant systems, and will focus on the state-space description of a so-called descriptor system. We line out the concept of shift invariance that occurs often in realization theory, and will turn out to be of great relevance for the task considered in this manuscript.

A.1 Linear Algebra Basics

A.1.1 Preliminaries

A *vector* is an 1D array of numbers and a *matrix* is a 2D array of numbers. In this manuscript, we usually consider vectors and matrices consisting of real numbers. However, in certain situations vectors or matrices may consist of complex numbers, therefore we present the general theory for the complex case, where necessary.

By default, vectors are assumed to be column vectors. We denote a vector as a bold-face lowercase latin symbol, *e.g.*, $\boldsymbol{b} \in \mathbb{C}^m$ is an m-dimensional vector (*i.e.,* $m \times 1$) consisting of complex numbers. Matrices are represented as a boldface uppercase latin symbol, *e.g.,* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a matrix having m rows and n columns consisting of real numbers..

Let us now review the concepts of linear dependency, subspaces and bases for subspaces.

Definition A.1 (Linear Dependency)**.** A set of vectors $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ is said to be *linearly independent* if $\sum_{j=1}^{n} \alpha_j \boldsymbol{a}_j = \boldsymbol{0}$ implies that $\alpha_1 = \ldots = \alpha_n = 0$. If, on the other hand, a nontrivial linear combination of the $\boldsymbol{a}_i$ is zero, then we say that $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ is *linearly dependent.*

Definition A.2 (Subspace and Basis)**.** A *subspace* of $\mathbb{C}^m$ is a subset that is also a vector space. The set of all linear combinations of a given collection of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \in \mathbb{C}^m$ is a subspace, also referred to as the *span* of $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, *i.e.,*

$$\operatorname{span}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\} := \left\{ \sum_{j=1}^{n} \beta_j \boldsymbol{a}_j : \beta_j \in \mathbb{C} \right\}.$$

A *basis* $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$ for a subspace $\mathcal{S}$ has two properties. First, it is linearly independent, and, second, it spans the subspace, *i.e.,* for all $\boldsymbol{s} \in \mathcal{S}$ we have that $\boldsymbol{s} = \sum_{j=1}^{k} \alpha_j \boldsymbol{b}_j$. All bases for a subspace $\mathcal{S}$ have the same number of elements, which is called the *dimension*, and is denoted by $\dim(\mathcal{S})$.

With any matrix $m \times n$ matrix $\boldsymbol{A}$ one can associate two important subspaces, namely the range and the null space.

Definition A.3. The *range* of the matrix $\boldsymbol{A}$ and the *null space* of the matrix $\boldsymbol{A}$ are defined as

$$\begin{aligned} \operatorname{range}(\boldsymbol{A}) &:= \{\boldsymbol{y} \in \mathbb{C}^m : \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \forall \boldsymbol{x} \in \mathbb{C}^n\}, \\ \operatorname{null}(\boldsymbol{A}) &:= \{\boldsymbol{x} \in \mathbb{C}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}. \end{aligned}$$

Let the column partitioning of $\boldsymbol{A}$ be denoted by

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ a_1 & a_2 & \ldots & a_n \\ | & | & & | \end{pmatrix},$$

then $\operatorname{range}(\boldsymbol{A}) = \operatorname{span}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$.

Next we come to a very important property, namely the rank of a matrix.

Definition A.4. The *rank* of the matrix $\boldsymbol{A}$ is defined as the dimension of its range, or formally

$$\operatorname{rank}(\boldsymbol{A}) := \dim(\operatorname{range}(\boldsymbol{A})).$$

In order to introduce the four fundamental subspaces of a matrix, it is instrumental to use real matrices.[1] A real matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has the following four interesting subspaces.

1. The *column space* of $\boldsymbol{A}$ is defined as range$(\boldsymbol{A})$.
2. The *row space* of $\boldsymbol{A}$ is defined as range$(\boldsymbol{A}^T)$.
3. The *null space* of $\boldsymbol{A}$ is null$(\boldsymbol{A})$ as defined above.
4. The *left null space* of $\boldsymbol{A}$ is null$(\boldsymbol{A}^T)$.

Recall that the dimension of the column space is called the rank of $\boldsymbol{A}$. A well-known property of matrices is that the rank of $\boldsymbol{A}$ is equal to the rank of $\boldsymbol{A}^T$, *i.e.,* the row rank and the column rank are equal: $\text{rank}(\boldsymbol{A}) = \text{rank}(\boldsymbol{A}^T) = r$. The dimension of the null space is called the *nullity*, *i.e.,*

$$\text{nullity}(\boldsymbol{A}) := \dim(\text{null}(\boldsymbol{A})).$$

The *rank-nullity theorem* describes a very useful relation between the rank and the nullity of a matrix.

Theorem A.5. For any $m \times n$ matrix $\boldsymbol{A}$, the following holds,

$$\text{nullity}(\boldsymbol{A}) + \text{rank}(\boldsymbol{A}) = n.$$

A consequence of the rank-nullity theorem is that the dimension of the left null space is $m - r$.

We now come to another important concept, namely the *inverse* of a matrix.

Definition A.6. Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be a square matrix. If we can find a matrix $\boldsymbol{X}$ for which $\boldsymbol{AX} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $n \times n$ identity matrix, we call $\boldsymbol{X}$ the inverse of $\boldsymbol{A}$, denoted as $\boldsymbol{A}^{-1}$.

If $\boldsymbol{A}^{-1}$ exists, $\boldsymbol{A}$ is said to be *nonsingular*; otherwise we say that $\boldsymbol{A}$ is said to be *singular*.

A *determinant* of a square matrix is a number that is a function of the entries of the matrix. Determinants take a central role in understanding the concepts of linear algebra and algebraic geometry, *e.g.,* when an explicit expression of the inverse of a matrix is required, or to study resultants. Although they are often cumbersome in a numerical linear algebra setting, we cannot avoid to define them in order to explain all relevant concepts. In the algorithms, however, we will avoid their usage.

[1] Similar considerations hold for complex matrices, but one needs to consider complex conjugated transpose $(\cdot)^*$. Notice that the interpretation as row space does not hold anymore in this case.

Laplace's formula allows to compute the determinant using the so-called minors. Let $\boldsymbol{A}$ be an $m \times m$ square matrix, of which the elements are denoted $a_{i,j}$. The minor $\boldsymbol{M}_{i,j}$ is defined as the determinant of the $(m-1) \times (m-1)$ matrix that results from $\boldsymbol{A}$ by removing the i-th row and the j-th column. The expression $(-1)^{i+j}\boldsymbol{M}_{i,j}$ is called a cofactor. The determinant of $\boldsymbol{A}$ is then given by the formula

$$\det(\boldsymbol{A}) = \sum_{i=1}^{m} (-1)^{i+j} a_{i,j} \boldsymbol{M}_{i,j}.$$

Furthermore, for a scalar a we have that $\det(a) = a$.

Example A.7. Consider the 3×3 matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 & 5 \\ -4 & 6 & 0 \\ 3 & 1 & -2 \end{pmatrix}.$$

Applying Laplace's formula for the third column, *i.e.*, $j = 3$ gives

$$\det(\boldsymbol{A}) = (-1)^{1+3} \cdot 5 \cdot \begin{vmatrix} -4 & 6 \\ 3 & 1 \end{vmatrix} + 0 + (-1)^{3+3} \cdot -2 \cdot \begin{vmatrix} 1 & 2 \\ -4 & 6 \end{vmatrix}.$$

The 2×2 determinants can be obtained similarly as

$$\begin{vmatrix} -4 & 6 \\ 3 & 1 \end{vmatrix} = (-1)^{1+1} \cdot -4 \cdot 1 + (-1)^{1+2} \cdot 6 \cdot 3 = -22,$$

$$\begin{vmatrix} 1 & 2 \\ -4 & 6 \end{vmatrix} = (-1)^{1+1} \cdot 1 \cdot 6 + (-1)^{1+2} \cdot 2 \cdot -4 = 14.$$

Finally we find $\det(\boldsymbol{A}) = -138$.

A *vector norm* is a function of a vector assigning a (positive) 'length' to a vector. In this manuscript the following two norms will be used:

- The Euclidean norm or 2-norm of a vector $\boldsymbol{v}$ of length n is defined as

  $$\|\boldsymbol{v}\|_2 := \sqrt{|v_1|^2 + \ldots + |v_n|^2}.$$

  Notice that $\|\boldsymbol{v}\|_2 = \boldsymbol{v}^* \boldsymbol{v}$.

- The Manhattan norm or 1-norm of a vector $\boldsymbol{v}$ of length n is defined as

  $$\|\boldsymbol{v}\|_1 := \sum_{i=1}^{n} |v_i|.$$

The notion of norm can be naturally generalized to matrices. We will use the following matrix norms:

- We will consider the operator norm, defined as

  $$\|A\| := \max\{\|Av\| : \|v\| = 1\}.$$

  When the 2-norm is considered, it can be proven that for a given $m \times n$ matrix A this leads to

  $$\begin{aligned}\|A\|_2 &:= \max\{\|Av\|_2 : \|v\| = 1\} \\ &= \sigma_1(A),\end{aligned}$$

  where $\sigma_1(A)$ denotes the first (largest) singular value of A.

- The Frobenius norm is defined as

  $$\begin{aligned}\|A\|_F &:= \sqrt{\textstyle\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \\ &= \sqrt{\textstyle\sum_{i=1}^{\min(m,n)}\sigma_i^2}.\end{aligned}$$

A.1.2 Eigenvalue Decomposition: Diagonalizable versus Non-diagonalizable Matrices

To understand the action of a matrix, it is often useful to bring a matrix into a more structured form. For a diagonal matrix, for instance, it is easy to understand what the action of a matrix is.

In the current section we will describe the eigenvalue decomposition, which expresses a matrix by means of its eigenvalues and eigenvectors.

Definition A.8 (Similarity Transform)**.** The $m \times m$ matrices A and B are called *similar* iff

$$B = P^{-1}AP.$$

Definition A.9 (Diagonalizable Matrix)**.** A matrix is called *diagonalizable* iff it is similar to a diagonal matrix. Alternatively, $A \in \mathbb{C}^{m\times m}$ is diagonalizable iff

$$B = P^{-1}AP,$$

where B is a diagonal matrix.

Eigenvalue Decomposition

Let $A \in \mathbb{C}^{m\times m}$ be a square matrix. We call the nonzero vector $v \in \mathbb{C}^m$ an *eigenvector* of A and λ its corresponding *eigenvalue* if

$$Av = \lambda v.$$

The eigenvalue decomposition of the matrix $\boldsymbol{A}$ is written as

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1},$$

where $\boldsymbol{V}$ is a matrix containing the eigenvectors $\boldsymbol{v}_j$ and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$. The eigenvalue decomposition can be rewritten as

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda},$$

or

$$\boldsymbol{A}\begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_m \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_m \\ | & & | \end{pmatrix}\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix}.$$

It is well-known that if the equation $\boldsymbol{M}\boldsymbol{v} = 0$ has a nontrivial solution, then $\det(\boldsymbol{M}) = 0$. Hence, the eigenvalues of $\boldsymbol{A}$ are the roots of its so-called characteristic polynomial $p(\lambda)$, defined as

$$p(\lambda) := \det(\boldsymbol{A} - \lambda\boldsymbol{I}).$$

A useful property is that a similarity transform leaves the eigenvalues of a matrix unchanged, whereas the eigenvectors are changed according to the change of basis prescribed by the similarity transform. Consider the $m \times m$ matrix $\boldsymbol{A}$. Let $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$ be its eigenvalue decomposition as defined above, and let λ denote an eigenvalue of $\boldsymbol{A}$ and let $\boldsymbol{v}$ be the corresponding eigenvector. We also consider the matrix $\boldsymbol{B} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$, defining the similarity transform. We then have that $\boldsymbol{B} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P} = (\boldsymbol{P}^{-1}\boldsymbol{V})\boldsymbol{\Lambda}(\boldsymbol{V}^{-1}\boldsymbol{P}) = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$, where $\boldsymbol{Q} = \boldsymbol{P}^{-1}\boldsymbol{V}$. We see from this that the eigenvalues of $\boldsymbol{A}$ are unchanged under the similarity transform. The eigenvectors of $\boldsymbol{B}$ are transformed according to the change of basis described by $\boldsymbol{P}$, *i.e.*, $\boldsymbol{w} = \boldsymbol{P}^{-1}\boldsymbol{v}$ is an eigenvector of $\boldsymbol{B}$ and λ is the corresponding eigenvalue.

Geometric and Algebraic Multiplicity

Not all square matrices are diagonalizable. A given $m \times m$ matrix $\boldsymbol{A}$ is not diagonalizable iff $\boldsymbol{A}$ has m linearly independent eigenvectors. This will lead us to the concepts of geometric and algebraic multiplicity of eigenvalues.

A matrix may have certain eigenvalues that occur several times. Such eigenvalues are called multiple eigenvalues. Consider an $m \times m$ matrix $\boldsymbol{A}$. We say that an eigenvalue λ_i has *algebraic multiplicity* $\mu(\lambda_i)$ if $(\lambda - \lambda_i)^{\mu(\lambda_i)}$ divides the characteristic polynomial $p(\lambda)$. The *geometric multiplicity* $\gamma(\lambda_i)$ of an eigenvalue λ_i is the dimension of the eigenspace corresponding to the eigenvalues λ_i, *i.e.*, the number of linearly independent eigenvectors corresponding to that eigenvalue.

It can be shown that the geometric multiplicity $\gamma(\lambda_i)$ can never exceed the algebraic multiplicity $\mu(\lambda_i)$, *i.e.*, $\gamma(\lambda_i) \leq \mu(\lambda_i), \forall i$.

Example A.10. For example, consider the 4×4 matrix

$$\boldsymbol{A} = \begin{pmatrix} -26 & 8 & 32 & 22 \\ 27 & -6 & -31 & -21 \\ -13 & 4 & 17 & 10 \\ -29 & 8 & 33 & 25 \end{pmatrix}.$$

The characteristic polynomial is $p(\lambda) = \det(\boldsymbol{A} - \lambda \boldsymbol{I}) = (\lambda - 2)^2(\lambda - 3)^2$. We have as the eigenvalues 2 and 3, both of which occur with (algebraic) multiplicity 2. The eigenvalue 2 has geometric multiplicity 2, since $\begin{pmatrix} .23 & -.61 & .63 & -.40 \end{pmatrix}^T$ and $\begin{pmatrix} -.27 & -.64 & .34 & -.62 \end{pmatrix}^T$ are the eigenvectors corresponding to the eigenvalue 2. The eigenvalue 3 has geometric multiplicity 1 since $\begin{pmatrix} -.53 & .26 & .00 & -.80 \end{pmatrix}^T$ is the only eigenvector corresponding to the eigenvalue 3. It is clear that the matrix $\boldsymbol{A}$ cannot be diagonalized by means of an eigenvalue decomposition.

Jordan Canonical Form

Although not all matrices are diagonalizable, in general, any square complex matrix can be decomposed into a block-diagonal form

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{J}\boldsymbol{P}^{-1},$$

where $\boldsymbol{J} = \text{diag}(\boldsymbol{J}_1, \boldsymbol{J}_2, \ldots, \boldsymbol{J}_q)$, and where the so-called Jordan block $\boldsymbol{J}_i$ defined as

$$\boldsymbol{J}_i := \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

is associated to the eigenvalue λ_i and has size m_i (with $m = \sum_{i=1}^{q} m_i$). (Note that the case where the matrix $\boldsymbol{A}$ is diagonalizable is a special case of m Jordan blocks of size $m_i = 1$).

Consider the equation $\boldsymbol{A}\boldsymbol{P} = \boldsymbol{J}\boldsymbol{P}$ and let $\boldsymbol{P} := \begin{pmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \ldots & \boldsymbol{P}_q \end{pmatrix}$, where $\boldsymbol{P}_i$ is composed of the columns of $\boldsymbol{P}$ associated with the i-th Jordan block $\boldsymbol{J}_i$. We then write

$$\boldsymbol{A}\boldsymbol{P}_i = \boldsymbol{P}_i\boldsymbol{J}_i,$$

where

$$\boldsymbol{P}_i := \begin{pmatrix} | & | & & | \\ v_{i,1} & v_{i,2} & \cdots & v_{i,n_i} \\ | & | & & | \end{pmatrix}.$$

Then we have that

$$\boldsymbol{A}\boldsymbol{v}_{i,1} = \lambda_i \boldsymbol{v}_{i,1},$$

so the first column of $\boldsymbol{P}_i$ is an eigenvector associated with the eigenvalue λ_i. For $j = 2, 3, \ldots, n_i$ we have that $\boldsymbol{A}\boldsymbol{v}_{i,j} = \boldsymbol{v}_{i,j-1} + \lambda_i \boldsymbol{v}_{i,j}$, defining the so-called *Jordan chain*. The vectors $\boldsymbol{v}_{i,j}$ are sometimes called the generalized eigenvectors.

Cayley-Hamilton Theorem

Let $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k$ be a polynomial and let $\boldsymbol{A}$ be an $m \times m$ matrix. We then define

$$p(\boldsymbol{A}) := a_0 \boldsymbol{I} + a_1 \boldsymbol{A} + a_2 \boldsymbol{A}^2 + \ldots + a_k \boldsymbol{A}^k.$$

In its simplest form, the Cayley-Hamilton theorem states that a matrix fulfills its own characteristic polynomial.

Theorem A.11 (Cayley-Hamilton Theorem)**.** Let $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ and let $p(\lambda) = \det(\boldsymbol{A} - \lambda \boldsymbol{I})$. Then we have that $p(\boldsymbol{A}) = 0$.

Corollary A.12. An important corollary of the Cayley-Hamilton theorem is that for every $n \geq m$ we can write $\boldsymbol{A}^n$ as a linear combination of the $\boldsymbol{A}^i$, with $i = 0, 1, \ldots, m$.

A.1.3 Two Important Decompositions: QR and SVD

QR Decomposition

In general, any $m \times n$ matrix can be decomposed into the product of an unitary matrix and an upper-triangular matrix. This is called the QR decomposition.

Theorem A.13. Any rectangular complex matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ with $m \geq n$ can be decomposed as

$$\boldsymbol{A} = \begin{array}{cc} & \begin{array}{cc} n & m-n \end{array} \\ m & \left(\begin{array}{cc} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{array} \right) \end{array} \begin{array}{cc} n & \\ \left(\begin{array}{c} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{array} \right) & \begin{array}{c} n \\ m-n \end{array} \end{array}$$

where $\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{pmatrix}$ is a unitary matrix (*i.e.*, $\boldsymbol{Q}^*\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^* = \boldsymbol{I}_m$) and $\boldsymbol{R}_1 \in \mathbb{C}^{n\times n}$ is upper-triangular. We have $\boldsymbol{Q}_1 \in \mathbb{C}^{m\times n}$ and $\boldsymbol{Q}_2 \in \mathbb{C}^{m\times m-n}$.

The QR decomposition is usually computed using well-conditioned Householder reflections or Givens rotations (Golub and Van Loan, 1996).

Singular Value Decomposition

The singular value decomposition (SVD) of a matrix is sometimes called the swiss army knife of numerical analysis. It shows up in a multitude of applied mathematics techniques and is the cornerstone of many mathematical modeling and analysis approaches. In this manuscript it will be used for

- determining the (numerical) rank of a matrix;
- finding a basis for the null space of a matrix;
- computing the so-called pseudo-inverse of a matrix; and,
- performing a column compression on a matrix.

Theorem A.14 (Singular Value Decomposition)**.** The singular value decomposition (SVD) of a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ with $m \geq n$ is

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*,$$

where $\boldsymbol{U} \in \mathbb{C}^{m \times m}$ and $\boldsymbol{V} \in \mathbb{C}^{n \times n}$ are unitary (*i.e.*, $\boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^* = \boldsymbol{I}_m$ and $\boldsymbol{V}^*\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^* = \boldsymbol{I}_n$). Furthermore, the matrix $\boldsymbol{\Sigma}$ is an $m \times n$ real matrix having the form

$$\boldsymbol{\Sigma} = \begin{array}{cc} \begin{array}{cc} r & n-r \end{array} & \\ \left(\begin{array}{cc} \boldsymbol{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right) & \begin{array}{c} r \\ m-r \end{array} \end{array} \tag{A.1}$$

where $r = \text{rank}(\boldsymbol{A})$, $\boldsymbol{\Sigma}_r := \text{diag}(\sigma_1, \ldots, \sigma_r)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ are called the singular values of $\boldsymbol{A}$.[2] The SVD exists for any given $m \times n$ matrix.

The SVD is a matrix decomposition method that can be applied to any square matrix, whereas the eigenvalue decomposition can only be applied to a square matrix. Nevertheless, the SVD bears a strong resemblance to the eigenvalue decomposition.

The columns of $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^*$ and the columns of $\boldsymbol{V}$ are the eigenvectors of $\boldsymbol{A}^*\boldsymbol{A}$. The r singular values on the diagonal of $\boldsymbol{\Sigma}$ are the square roots of the nonzero eigenvalues of both $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$.

The questions we consider in this thesis are typically studied in the mathematical discipline called algebraic geometry. It is in many cases instructive to consider the geometrical interpretation of an algebraic notion. The SVD has

[2] The rank of the matrix is read off as the number of nonzero singular values. The SVD is the most reliable method for determining the (numerical) rank of a matrix, which is in practice done by counting the number of singular values that are greater than a certain user-defined threshold value. The default threshold value used in MATLAB is $\max(m,n) \cdot \epsilon(\sigma_1)$ where $\epsilon(\sigma_1)$ is the distance from σ_1 to the next larger in magnitude floating point number of the same precision as σ_1.

an interesting geometric interpretation. Let us therefore consider the SVD of a real matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Partitioning $\boldsymbol{U}$ and $\boldsymbol{V}$ accordingly with the partitioning of $\boldsymbol{\Sigma}$ as in (A.1) results in

$$\boldsymbol{A} = \begin{array}{cc} & \begin{array}{cc} r & m-r \end{array} \\ m & \begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cc} r & n-r \end{array} \\ \begin{pmatrix} \boldsymbol{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array} \begin{array}{cc} n & \\ \begin{pmatrix} \boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T \end{pmatrix} & \begin{array}{c} r \\ n-r \end{array} \end{array}$$

The SVD provides numerical bases for the four fundamental subspaces of a matrix, namely the column space, the row space, the left and right null space. We have

$$\begin{aligned} \operatorname{range}(\boldsymbol{U}_1) &= \operatorname{range}(\boldsymbol{A}), \\ \operatorname{range}(\boldsymbol{U}_2) &= \operatorname{null}(\boldsymbol{A}^T), \\ \operatorname{range}(\boldsymbol{V}_1) &= \operatorname{range}(\boldsymbol{A}^T), \\ \operatorname{range}(\boldsymbol{V}_2) &= \operatorname{null}(\boldsymbol{A}). \end{aligned}$$

Pseudo-inverse of a Matrix

The inverse of a rectangular matrix is not defined, however, a generalization of the inversion operator is given by the pseudo-inverse. Let $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ with $m \geq n$ and its SVD is given by $\boldsymbol{A} = \boldsymbol{U\Sigma V}^*$. The *Moore-Penrose pseudo-inverse of* $\boldsymbol{A}$ (or pseudo-inverse), denoted by $\boldsymbol{A}^+$, is defined as

$$\boldsymbol{A}^+ := \boldsymbol{V\Sigma}^+\boldsymbol{U}^*,$$

where $\boldsymbol{\Sigma}^+ := \operatorname{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_r, 0, \ldots, 0)$ has dimensions $n \times m$ and $r = \operatorname{rank}(\boldsymbol{A})$.

Row Compression and Column Compression

Due to the fact the use of orthogonal (unitary) transformations, the SVD can be used to numerically reliably perform certain transformations on a given matrix. We now describe the so-called row compression and column compression operations to 'compress' a matrix such that certain rows or columns, respectively, contain vectors that span the rank of the matrix.

Theorem A.15. Let the SVD of $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ be given by $\boldsymbol{A} = \boldsymbol{U\Sigma V}^*$. Then we have that

$$\begin{aligned} \boldsymbol{U}^*\boldsymbol{A} &= \boldsymbol{\Sigma V}^* \\ &= \begin{pmatrix} \boldsymbol{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_1^* \\ \boldsymbol{V}_2^* \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Sigma}_r \boldsymbol{V}_1^* \\ \mathbf{0} \end{pmatrix} \in \mathbb{C}^{m \times n} \end{aligned}$$

We have that null($\boldsymbol{A}$) = null($\boldsymbol{U}^*\boldsymbol{A}$) = null($\boldsymbol{\Sigma}_r\boldsymbol{V}_1^*$) and the matrix $\boldsymbol{\Sigma}_r\boldsymbol{V}_1^* \in \mathbb{C}^{r\times n}$ has full row rank. The pre-multiplication of $\boldsymbol{A}$ by $\boldsymbol{U}^*$ 'compresses' $\boldsymbol{A}$ by row transformations.

In the same way one can consider the column compression.

Theorem A.16. Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ have the SVD as $\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$. Then we have

$$\begin{aligned} \boldsymbol{AV} &= \boldsymbol{U\Sigma} \\ &= \begin{pmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{U}_1\boldsymbol{\Sigma}_r & \boldsymbol{0} \end{pmatrix} \in \mathbb{C}^{m\times n} \end{aligned}$$

Notice that range($\boldsymbol{A}$) = range($\boldsymbol{AV}$) = range($\boldsymbol{U}_1\boldsymbol{\Sigma}_r$) and the matrix $\boldsymbol{U}_1\boldsymbol{\Sigma}_r \in \mathbb{C}^{m\times r}$ has full column rank. The post-multiplication of $\boldsymbol{A}$ by $\boldsymbol{V}$ is a transformation that 'compresses' $\boldsymbol{A}$ by column transformations.

A.1.4 Projections and Least Squares

Orthogonal Projection

Consider a set of vectors $\boldsymbol{a}_i$, with $i = 1, \dots, n$ and a given vector $\boldsymbol{b} \notin \text{span}\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_n\}$. We assume that all vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}$ are consisting of real numbers. Let $\boldsymbol{A}$ denote the matrix having the vectors $\boldsymbol{a}_i$ as its columns. Let $\boldsymbol{p}$ denote the orthogonal projection of $\boldsymbol{b}$ onto the space spanned by the vectors $\boldsymbol{a}_i$, such that $\boldsymbol{p} = \boldsymbol{Ax} = \boldsymbol{b} - \boldsymbol{e}$ for some unknown vector $\boldsymbol{x}$, and $\boldsymbol{e}$ is the difference between $\boldsymbol{b}$ and its orthogonal projection onto the space spanned by the $\boldsymbol{a}_i$. We then have that $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{Ab}$, and $\boldsymbol{e} \perp \boldsymbol{a}_i, i = 1, \dots, n$. Hence we can write

$$\begin{aligned} \boldsymbol{a}_1^T\boldsymbol{e} &= 0, \\ \boldsymbol{a}_2^T\boldsymbol{e} &= 0, \\ &\vdots \\ \boldsymbol{a}_n^T\boldsymbol{e} &= 0, \end{aligned}$$

or, more compactly,

$$\boldsymbol{A}^T(\boldsymbol{b} - \boldsymbol{Ax}) = 0 \quad \Rightarrow \quad \boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}.$$

We can express $\boldsymbol{p}$ by means of the so-called projection matrix $\boldsymbol{P}$ defined by $\boldsymbol{p} = \boldsymbol{Pb}$, leading to

$$\boldsymbol{p} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{Pb}.$$

The residual $\boldsymbol{e}$ can be expressed as

$$\boldsymbol{e} = (\boldsymbol{I}_n - \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T)\boldsymbol{b}.$$

Least Squares

Today the least squares (LS) method, with its roots going back to Gauss (1809) and Legendre (1806), is still the basis for many modeling schemes in applied mathematics and engineering, in particular in a (parameter) estimation setting.

In its simplest form, *i.e.*, the linear least squares problem, this method can be viewed as the task of fitting a linear relation to observed noisy data — or alternatively, to approximately solve an overdetermined linear system (in practice, the amount of data usually ensures there are more equations than unknowns). Assume that the data $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$ are the independent variables and $\boldsymbol{b} \in \mathbb{R}^{m}$ are the dependent variables that are observed, between which linear relations are expected (however not exact due to measurement errors). We then have

$$\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}.$$

One now tries to correct the data $\boldsymbol{b}$ as little as possible such that the equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} + \boldsymbol{\Delta b} \tag{A.2}$$

holds. This 'minimal' correction to the data $\boldsymbol{b}$ is expressed mathematically as the minimization of the sum of squared residuals $\boldsymbol{\Delta b}$, subject to the model equation (A.2). If $\boldsymbol{A}$ is of full column rank n, the solution is found from the so-called normal equations

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}.$$

In the case the errors are normally distributed ($\boldsymbol{\Delta b} \sim \mathcal{N}(0,\sigma I)$), the least squares estimator corresponds with the Maximum Likelihood estimator. Observe that a low-rank matrix approximation problem underlies this task, which becomes clear by investigating (A.2):

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{b} + \boldsymbol{\Delta b} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ -1 \end{pmatrix} = \boldsymbol{0}_m.$$

It turns out that behind the scenes, a low-rank matrix approximation problem is the core task in many (linear) modeling tasks, such as linear dynamical system identification.

Total Least Squares

A well-known extension of the LS method, is the total least squares (TLS) problem, also known as errors-in-variables, or orthogonal regression. This problem is encountered when both the independent variables $\boldsymbol{A}$ and the dependent variables $\boldsymbol{b}$ are subject to measurement errors. Both $\boldsymbol{A}$ and $\boldsymbol{b}$ are adjusted as little as possible such that the corrected data are related linearly, or equivalently

$$(\mathbf{A} + \Delta\mathbf{A})\mathbf{v} = \mathbf{b} + \Delta\mathbf{b}, \tag{A.3}$$

which corresponds to the optimization problem

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & \left\| \begin{pmatrix} \boldsymbol{\Delta A} & \boldsymbol{\Delta b} \end{pmatrix} \right\|_F^2, \\ \text{subject to} \quad & (\boldsymbol{A} + \boldsymbol{\Delta A})\boldsymbol{x} = \boldsymbol{b} + \boldsymbol{\Delta b}, \\ & \boldsymbol{x}^T\boldsymbol{x} = 1. \end{aligned}$$

Again, there is an obvious underlying low-rank matrix approximation problem at work, which one can understand from(A.3). The computational task of finding a solution to the TLS problem is done by finding the SVD of the augmented data matrix $\boldsymbol{M} := \begin{pmatrix} \boldsymbol{A} & \boldsymbol{b} \end{pmatrix}$. The SVD equations are given by

$$\begin{cases} \boldsymbol{M}\mathbf{v} & = & \boldsymbol{u}\sigma, \\ \boldsymbol{M}^T\mathbf{u} & = & \boldsymbol{v}\sigma, \\ \boldsymbol{v}^T\mathbf{v} & = & 1, \\ \boldsymbol{u}^T\mathbf{u} & = & 1. \end{cases} \tag{A.4}$$

Since $\left\| \begin{pmatrix} \boldsymbol{\Delta A} & \boldsymbol{\Delta b} \end{pmatrix} \right\|_F^2 = \sigma^2$, one needs to find the singular triplet corresponding to the smallest singular value. Also in this case, the TLS estimator corresponds to the Maximum Likelihood estimator when one assumes i.i.d. additive Gaussian noise: $\begin{pmatrix} \Delta A & \boldsymbol{\Delta b} \end{pmatrix} \sim \mathcal{N}(0, \sigma I)$. Again, it is natural to view this scheme as a low-rank matrix approximation method, and the underlying low-rank matrix may in this case be reconstructed as $\boldsymbol{M} - \boldsymbol{u}\sigma\boldsymbol{v}^T$, hence a rank-one modification of $\boldsymbol{M}$.

The approximation in Frobenius norm of a given matrix by one of lower rank has a rich and long history, with its roots tracing back to Adcock (1877, 1878), over Pearson (1901), Eckart and Young (1936), Householder and Young (1938), to Golub and Van Loan (1980, 1996). Several geometric, algebraic and statistical details can be found in Golub and Van Loan (1996) and Van Huffel and Vandewalle (1991).

Structured Total Least Squares

When additionally to the former case, the preservation of a specific matrix structure, and/or a weighting of specific matrix entries has to be taken into account, the resulting approximation task is termed the Structured, respectively the Weighted Total Least Squares problem (STLS, respectively WTLS). Useful references are De Moor (1993, 1994b); Lemmerling (1999); Markovsky (2008); Rosen et al. (1996).

Consider an affinely structured data matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{m \times n}$ of full column rank n, which is to be approximated by a low-rank matrix $\boldsymbol{B} = (b_{ij}) \in \mathbb{R}^{m \times n}$ such that the affine structure is preserved.[3] Moreover, an element-wise weighting $\boldsymbol{W} = (w_{ij}) \in \mathbb{R}^{m \times n}$ can be taken into account. This can be cast as an optimization problem in the following way

$$\begin{array}{ll} \underset{\boldsymbol{B},\boldsymbol{v}}{\text{minimize}} & \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} - b_{ij})^2 w_{ij}, \\ \\ \text{subject to} & \boldsymbol{B}\boldsymbol{v} = 0, \\ & \boldsymbol{v}^T\boldsymbol{v} = 1, \\ & \boldsymbol{B} \text{ structured.} \end{array}$$

Two major observations form the underlying motivation for the introduction of this matrix approximation scheme:

1. It turns out that matrices that are simultaneously affinely structured and rank deficient occur frequently in many signal processing and system identification applications. In these applications, the combination of rank deficiency and structure implies that the underlying data are 'generated' by a dynamical system which is linear-in-the-parameters. However, when dealing with real-life applications, observations are usually corrupted by additive (measurement) errors, and therefore, the observed data matrices are never exactly rank deficient. The STLS problem provides a scheme to correct the data in a least squares sense, such that an approximation of the underlying system can be reconstructed.

2. One can tackle this constrained optimization problem by applying the Lagrange multipliers method to the constrained optimization problem. When doing so, a 'nonlinear' generalized SVD is found: whereas the SVD arose as an elegant formulation to solve the TLS problem, its counterpart for STLS was termed the Riemannian SVD in De Moor (1994a), and is given by the following equations

$$\left\{ \begin{array}{rcl} \boldsymbol{M}\boldsymbol{v} & = & \boldsymbol{D}_v \boldsymbol{u} \tau, \\ \boldsymbol{M}^T\boldsymbol{u} & = & \boldsymbol{D}_u \boldsymbol{v} \tau, \\ \boldsymbol{v}^T\boldsymbol{v} & = & 1, \\ \boldsymbol{u}^T\boldsymbol{D}_v\boldsymbol{u} & = & 1\,(= \boldsymbol{v}^T\boldsymbol{D}_u\boldsymbol{v}), \end{array} \right. \tag{A.5}$$

   where $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ are a left, respectively right singular vector, and $\tau \in \mathbb{R}$ is the corresponding singular value. Notice the resemblance

[3] Affinely structured matrices $\boldsymbol{M}$ can be written as an affine (linear) combination of a given set $\{\boldsymbol{M}_k; k = 0, 1, \ldots, s\}$ of $s + 1$ basis matrices as $\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{M}_1 m_1 + \boldsymbol{M}_2 m_2 + \ldots + \boldsymbol{M}_s m_s$. Here, the real coefficients $m_k, k = 1, 2, \ldots, s$ 'parametrize' the structured matrix $\boldsymbol{M}$. Examples of such matrices are symmetric matrices, (block) Hankel, (block) Toeplitz, matrices with a certain sparsity pattern, *etc.*

with the SVD, Equations A.4. The important difference between the SVD and the Riemannian SVD lies in the symmetric non-negative definite matrices $\boldsymbol{D}_v$ and $\boldsymbol{D}_u$, the elements of which are quadratic functions of the components of $\boldsymbol{u}$, respectively $\boldsymbol{v}$. Their precise structure depends on the matrix structure and on the weights in the objective function.

Again, the low-rank approximation problem proceeds by seeking the minimal singular value τ: it can be shown that τ is exactly equal to the objective function, *i.e.*,

$$\tau = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - b_{ij})^2 w_{ij}.$$

However, when solving the Riemannian SVD, one encounters many suboptimal local solutions due to the structure constraint. A heuristic approach to tackle this problem has been presented in De Moor (1993): an iterative alternating scheme is applied and in consecutive steps, 1) the Riemannian SVD is solved as a generalized SVD by fixing $\boldsymbol{u}$ and $\boldsymbol{v}$ (and hence also $\boldsymbol{D}_u$ and $\boldsymbol{D}_v$, and next, 2) an update of $\boldsymbol{u}$ and $\boldsymbol{v}$ is computed and the algorithm returns to step 1) until convergence. In the specific implementation, QR matrix decompositions are employed to speed up the computations.

A.2 State-space Model for 1D Systems

A dynamical linear time-invariant (LTI) discrete time system can be described using the so-called state-space description

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k), \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k), \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{A.6}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^\theta$ denotes the state vector at time instant k, $\boldsymbol{u}(k) \in \mathbb{R}^\rho$ denotes the input at time instant k, and $\boldsymbol{y}(k) \in \mathbb{R}^\phi$ is the output vector at time instant k. The initial state is given as $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Iterating the system equations, starting from a given initial state $\boldsymbol{x}_0$ and a sequence of inputs $\boldsymbol{u}(k)$ for $k = 0, 1, \ldots$, leads to

$$\begin{aligned} \boldsymbol{x}(1) &= \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{B}\boldsymbol{u}(0), \\ \boldsymbol{y}(0) &= \boldsymbol{C}\boldsymbol{x}_0 + \boldsymbol{D}\boldsymbol{u}(0), \\ \\ \boldsymbol{x}(2) &= \boldsymbol{A}^2\boldsymbol{x}_0 + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(0) + \boldsymbol{B}\boldsymbol{u}(1), \\ \boldsymbol{y}(1) &= \boldsymbol{C}\boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}(0) + \boldsymbol{D}\boldsymbol{u}(1), \\ \\ \boldsymbol{x}(3) &= \boldsymbol{A}^3\boldsymbol{x}_0 + \boldsymbol{A}^2\boldsymbol{B}\boldsymbol{u}(0) + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(1) + \boldsymbol{B}\boldsymbol{u}(2), \\ \boldsymbol{y}(2) &= \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{x}_0 + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\boldsymbol{u}(0) + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}(1) + \boldsymbol{D}\boldsymbol{u}(2), \end{aligned}$$

In general we obtain

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}^{k+1}\boldsymbol{x}(0) + \boldsymbol{A}^{k}\boldsymbol{B}\boldsymbol{u}(0) + \boldsymbol{A}^{k-1}\boldsymbol{B}\boldsymbol{u}(1) + \ldots + \boldsymbol{B}\boldsymbol{u}(k), \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{A}^{k}\boldsymbol{x}(0) + \boldsymbol{C}\boldsymbol{A}^{k-1}\boldsymbol{B}\boldsymbol{u}(0) + \ldots + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}(k-1) + \boldsymbol{D}\boldsymbol{u}(k). \end{aligned} \tag{A.7}$$

A.3 Computing the Output to a Given Input Signal

A.3.1 SISO Systems

A fundamental result in LTI system theory is that a system can be completely characterized by its so-called *impulse response*. Let us explain this by starting with a single-input single-output (SISO) system. When dealing with a SISO system, the input and output are scalar signals, *i.e.*, $\rho = \phi = 1$, hence we denote the input and output by u and y, respectively. Furthermore the SISO state-space description is simplified to

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{b}u, \\ y(k) &= \boldsymbol{c}^{T}\boldsymbol{x}(k) + du, \end{aligned}$$

with $\boldsymbol{x} \in \mathbb{R}^{\theta}$, $\boldsymbol{A} \in \mathbb{R}^{\theta\times\theta}$, $\boldsymbol{b} \in \mathbb{R}^{\rho}$, $\boldsymbol{c} \in \mathbb{R}^{\phi}$, $u(k) \in \mathbb{R}$, $y(k) \in \mathbb{R}$, and $d \in \mathbb{R}$.

The impulse response of a SISO system is the sequence of outputs $y(k)$ obtained by applying the *impulse input signal* $\delta(k)$ defined as

$$\delta(k) = \begin{cases} 1 & \text{if} \quad k = 0, \\ 0 & \text{if} \quad k \neq 0. \end{cases} \tag{A.8}$$

It can be seen from (A.7) that the output of the system, which is called the impulse response, denoted by $g(k)$, is then given by

$$g(k) = \begin{cases} d & \text{if} \quad k = 0, \\ \boldsymbol{c}^{T}\boldsymbol{A}^{k-1}\boldsymbol{b} & \text{if} \quad k > 0. \end{cases}$$

The output of a SISO system to any given input is simply the convolution of the input with the system's impulse response. The convolution of two (univariate) signals, $u(k)$ and $g(k)$ is defined as $(u * g)(k) := \sum_{m=-\infty}^{\infty} u(m)g(k-m)$, however the same operation can be expressed starting from (A.7) as follows. We have

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{i-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}^{T} \\ \boldsymbol{c}^{T}\boldsymbol{A} \\ \boldsymbol{c}^{T}\boldsymbol{A}^{2} \\ \vdots \\ \boldsymbol{c}^{T}\boldsymbol{A}^{i-1} \end{pmatrix} \boldsymbol{x}_0 + \boldsymbol{T}_i \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{i-1} \end{pmatrix},$$

where the impulse response samples $g(k)$ are placed in a Toeplitz matrix $\boldsymbol{T}_i$, where

$$\boldsymbol{T}_i := \begin{pmatrix} g(0) & 0 & 0 & \cdots & 0 \\ g(1) & g(0) & 0 & \cdots & 0 \\ g(2) & g(1) & g(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ g(i-2) & g(i-3) & g(i-4) & \cdots & 0 \\ g(i-1) & g(i-2) & g(i-3) & \cdots & g(0) \end{pmatrix}.$$

A.3.2 MIMO Systems

Impulse response experiments of a multiple-input multiple-output (MIMO) system are a natural generalization of the above-mentioned ideas. In accordance with the SISO case again we can derive from (A.7) the responses to an input signal $\boldsymbol{u}(k)$ that is 0 except for at $k = 0$ as

$$\boldsymbol{G}(k) = \begin{cases} \boldsymbol{D} & \text{if} \quad k = 0, \\ \boldsymbol{C}\boldsymbol{A}^{k-1}\boldsymbol{B} & \text{if} \quad k > 0. \end{cases} \tag{A.9}$$

These matrices are called the *Markov parameters*. In practice the MIMO impulse response experiment is executed as follows. First, an impulse signal $\delta(k)$ is applied to the first input, while applying a zero signal to all other inputs. The response of this experiment provides us with a sequence of vectors that will compose the first columns of the Markov parameters matrices. Next, an impulse signal (A.8) is applied to the second input, while applying a zero signal to all other inputs. The outcome of this experiment will give us the second columns, *etc.*

A.4 Realization Theory for 1D Systems

A.4.1 Regular 1D Systems

We consider a state-space model of the form (A.6). An important problem for this manuscript is the so-called realization problem, which aims at finding a system description from a given set of impulse response matrices.

Problem A.17 (Realization Problem)**.** Given a set of impulse response matrices $\boldsymbol{G}(k)$, for $k = 0, \ldots, N-1$, find the state dimension θ and a system realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$.

By embedding the observed impulse response into an appropriately sized Hankel matrix, the essential information about the underlying model is revealed, *e.g.*, the rank of the constructed Hankel matrix corresponds to the McMillan degree of the underlying linear dynamical system — and moreover, a state-space model (the so-called *realization*) is estimated from a decomposition of this Hankel data matrix.

The extended observability matrix $\mathcal{O}_i$ $(i > \theta)$ is defined as

$$\mathcal{O}_i = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^{i-1} \end{pmatrix}, \tag{A.10}$$

and the extended controllability matrix $\mathcal{C}_j$ $(j > n)$ is defined as

$$\mathcal{C}_j = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{AB} & \dots & \boldsymbol{A}^{j-1}\boldsymbol{B} \end{pmatrix}. \tag{A.11}$$

Consider the $i\phi \times j\rho$ block Hankel matrix $\boldsymbol{H}_{i,j}$ with block dimensions i and j and $(i+j-1) \leq K$, which is defined as

$$\boldsymbol{H}_{i,j} = \begin{pmatrix} \boldsymbol{G}(1) & \boldsymbol{G}(2) & \boldsymbol{G}(3) & \dots & \boldsymbol{G}(j) \\ \boldsymbol{G}(2) & \boldsymbol{G}(3) & \boldsymbol{G}(4) & \dots & \boldsymbol{G}(j+1) \\ \boldsymbol{G}(3) & \boldsymbol{G}(4) & & \iddots & \vdots \\ \vdots & \vdots & \iddots & & \vdots \\ \boldsymbol{G}(i) & \boldsymbol{G}(i+1) & \dots & \dots & \boldsymbol{G}(i+j-1) \end{pmatrix}. \tag{A.12}$$

An obvious consequence from Equation A.9 is that the block Hankel matrix $\boldsymbol{H}_{i,j}$ can be factorized into the product of the extended observability matrix and the extended controllability matrix, *i.e.*,

$$\boldsymbol{H}_{i,j} = \mathcal{O}_i \mathcal{C}_j.$$

If i and j are sufficiently large — which we will assume — this block Hankel matrix is rank deficient. Furthermore, its rank is equal to n, the McMillan degree of the system.

A boiled-down version of the realization algorithm of Ho and Kalman (1966) is given in Algorithm 7.

Algorithm 7. *(Realization Algorithm (Ho and Kalman, 1966))*

input: Markov parameters $\boldsymbol{G}(k), k = 0, \dots, K$
output: (Minimal order) realization $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$

1. The matrix $\boldsymbol{D}$ is easily found as

$$\boldsymbol{D} = \boldsymbol{G}(0). \tag{A.13}$$

2. Construct the (block-)Hankel matrix as in (A.12).

3. Perform an SVD on $\boldsymbol{H}_{i,j}$:

$$\boldsymbol{H}_{i,j} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

   and take

$$\begin{aligned} \mathcal{O}_i &= \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}, \\ \mathcal{C}_j &= \boldsymbol{\Sigma}^{1/2}\boldsymbol{V}^T. \end{aligned}$$

   The rank of the block Hankel matrix, the minimal order of the underlying system, is equal to the number of nonzero singular values.

4. $\boldsymbol{C}$ is formed from the first ϕ rows of $\mathcal{O}_i$, while $\boldsymbol{B}$ is formed from the first ρ columns of $\mathcal{C}_j$.

5. It follows from (A.10) that[4]

$$\underline{\mathcal{O}_i}\boldsymbol{A} = \overline{\mathcal{O}_i}, \tag{A.14}$$

   so $\boldsymbol{A}$ can be calculated as

$$\boldsymbol{A} = \underline{\mathcal{O}_i}^{+}\overline{\mathcal{O}_i}. \tag{A.15}$$

   Analogously, $\boldsymbol{A}$ can also be calculated from (A.11) as

$$\boldsymbol{A} = |\mathcal{C}_j \;\; \mathcal{C}_j|^{+}. \tag{A.16}$$

A.4.2 Realization Theory for 1D Descriptor Systems

Consider the autonomous linear time-invariant system

$$\begin{aligned} \widetilde{\boldsymbol{E}}\boldsymbol{x}(k+1) &= \widetilde{\boldsymbol{A}}\boldsymbol{x}(k) \\ \boldsymbol{y}(k) &= \widetilde{\boldsymbol{C}}\boldsymbol{x}(k), \end{aligned}$$

where $\boldsymbol{y}(k) \in \mathbb{R}^{\phi}$ and $\boldsymbol{x}(k) \in \mathbb{R}^{\theta}$ denote the output and state vector at time instant k. For identification purposes, the matrices $\widetilde{\boldsymbol{E}}$, $\widetilde{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{C}}$ are unknown matrices of appropriate dimensions. This system called a *descriptor system* if the matrix $\widetilde{\boldsymbol{E}}$ is singular, otherwise it is called a regular system.[5]

We assume that the pencil $\lambda\boldsymbol{E} - \widetilde{\boldsymbol{A}}$ is given in the Kronecker Canonical Form (KCF), meaning that $\widetilde{\boldsymbol{E}}$ and $\widetilde{\boldsymbol{A}}$ are block diagonal matrices, which bears no loss of generality.[6]

[4] If $\boldsymbol{X}$ is a $i\phi \times j\rho$ block matrix, with $\phi \times \rho$ matrices as its blocks, then $\overline{\boldsymbol{X}}$, $\underline{\boldsymbol{X}}$ are $(i-1)\phi \times j\rho$ matrices constructed from $\boldsymbol{X}$ by omission of the first, last block row, respectively. Similarly, $|\boldsymbol{X}$, $\boldsymbol{X}|$ are the $i\phi \times (j-1)\rho$ block matrices constructed from $\boldsymbol{X}$ by omission of its first, last block column, respectively.

[5] Note that, if $\widetilde{\boldsymbol{E}}$ is invertible, the equation $\widetilde{\boldsymbol{E}}\boldsymbol{x}(k+1) = \widetilde{\boldsymbol{A}}\boldsymbol{x}(k)$ can be rewritten as $\boldsymbol{x}(k+1) = \widetilde{\boldsymbol{E}}^{-1}\widetilde{\boldsymbol{A}}\boldsymbol{x}(k)$, resulting in a regular system as in (A.6).

[6] In the case the matrix pencil is regular, the Kronecker canonical form is also called the Weierstrass canonical form. A nice discussion about the Kronecker and Weierstrass canonical forms is beyond the scope of this text. In Gerdin (2004a,b) a nice introduction can be found, together with algorithms for computing them and several applications. The Kronecker and Weierstrass canonical forms are also discussed in Gantmacher (1960, Chapter 12). The original works regarding these canonical forms are Kronecker (1890) and Weierstrass (1868).

Consider therefore the equivalent model

$$\begin{aligned}\left(\boldsymbol{P}\widetilde{\boldsymbol{E}}\boldsymbol{Q}\right)\left(\boldsymbol{Q}^{-1}\boldsymbol{x}(k+1)\right) &= \left(\boldsymbol{P}\widetilde{\boldsymbol{A}}\boldsymbol{Q}\right)\left(\boldsymbol{Q}^{-1}\boldsymbol{x}(k)\right),\\ \boldsymbol{y}(k) &= \left(\widetilde{\boldsymbol{C}}\boldsymbol{Q}\right)\left(\boldsymbol{Q}^{-1}\boldsymbol{x}(k)\right),\end{aligned}$$

which allows to decompose the pencil $\lambda\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}$ into a regular and a singular part

$$\boldsymbol{P}\left(\lambda\widetilde{\boldsymbol{E}} - \widetilde{\boldsymbol{A}}\right)\boldsymbol{Q} = \left(\begin{array}{c|c} \boldsymbol{A}_R - \lambda\boldsymbol{I} & \boldsymbol{0} \\ \hline \boldsymbol{0} & \lambda\boldsymbol{E}_S - \boldsymbol{I}\end{array}\right),$$

where $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{\theta\times\theta}$.

The system equations can then be transformed into the reduced model (Moonen et al., 1992)

$$\begin{aligned}\left(\begin{array}{c} \boldsymbol{v}(k+1) \\ \hline \boldsymbol{w}(k-1)\end{array}\right) &= \left(\begin{array}{c|c} \boldsymbol{A}_R & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{E}_S\end{array}\right)\left(\begin{array}{c} \boldsymbol{v}(k) \\ \hline \boldsymbol{w}(k)\end{array}\right),\\ \boldsymbol{y}(k) &= \left(\begin{array}{c|c} \boldsymbol{C}_R & \boldsymbol{C}_S\end{array}\right)\left(\begin{array}{c} \boldsymbol{v}(k) \\ \hline \boldsymbol{w}(k)\end{array}\right),\end{aligned}$$

which separates the regular part (denoted by the state vector $\boldsymbol{v} \in \mathbb{R}^{\theta_R}$) and singular part (denoted by the state vector $\boldsymbol{w} \in \mathbb{R}^{\theta_S}$), with $\theta_R + \theta_S = \theta$. Notice that the singular part is implemented using an iteration running backward in time. Let the state vector sequences $\boldsymbol{V}_k$ and $\boldsymbol{W}_k$ be defined as[7]

$$\begin{aligned}\boldsymbol{V}_k &:= \left(\begin{array}{c|c|c|c} \boldsymbol{v}_k & \boldsymbol{v}_{k+1} & \ldots & \boldsymbol{v}_{k+j-1}\end{array}\right), \text{ and}\\ \boldsymbol{W}_k &:= \left(\begin{array}{c|c|c|c} \boldsymbol{w}_k & \boldsymbol{w}_{k+1} & \ldots & \boldsymbol{w}_{k+j-1}\end{array}\right).\end{aligned}$$

Define the output Hankel matrix $\boldsymbol{Y}_{1|i}$ as (with typically $j \gg i$)

$$\boldsymbol{Y}_{1|i} = \begin{pmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_j \\ \boldsymbol{y}_2 & \boldsymbol{y}_3 & \cdots & \boldsymbol{y}_{j+1} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{y}_i & \boldsymbol{y}_{i+1} & \cdots & \boldsymbol{y}_{i+j-1}\end{pmatrix}$$

We now have

$$\boldsymbol{Y}_{1|i} = \left(\begin{array}{c|c} \boldsymbol{C}_R & \boldsymbol{C}_S\boldsymbol{E}_S^{i-1} \\ \boldsymbol{C}_R\boldsymbol{A}_R & \vdots \\ \vdots & \boldsymbol{C}_S\boldsymbol{E}_S \\ \boldsymbol{C}_R\boldsymbol{A}_R^{i-1} & \boldsymbol{C}_S\end{array}\right)\left(\begin{array}{cccc} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots & \boldsymbol{v}_j \\ \hline \boldsymbol{w}_i & \boldsymbol{w}_{i+1} & \ldots & \boldsymbol{w}_{i+j-1}\end{array}\right)$$

Indeed, the following equation is a classical result in realization theory

$$\boldsymbol{Y}_{1|i} = \boldsymbol{\Gamma}\begin{pmatrix} \boldsymbol{V}_1 \\ \boldsymbol{W}_i\end{pmatrix},$$

[7] Remark that, for notational convenience, we sometimes employ the simplified notation $\boldsymbol{x}_k$ to denote $\boldsymbol{x}(k)$.

where

$$\mathbf{\Gamma} := \left(\begin{array}{c|c} \mathbf{\Gamma}_R & \mathbf{\Gamma}_C \end{array} \right) := \left(\begin{array}{c|c} \boldsymbol{C}_R & \boldsymbol{C}_S \boldsymbol{E}_S^{i-1} \\ \boldsymbol{C}_R \boldsymbol{A}_R & \vdots \\ \vdots & \boldsymbol{C}_S \boldsymbol{E}_S \\ \boldsymbol{C}_R \boldsymbol{A}_R^{i-1} & \boldsymbol{C}_S \end{array} \right) \tag{A.17}$$

Notice that the nilpotency of $\boldsymbol{E}_S$ prescribes the existence of a μ for which $\boldsymbol{E}_S^d = 0$ for $d \geq \mu$. Consequently, in $\mathbf{\Gamma}$ the following structure emerges,

$$\mathbf{\Gamma} = \left(\begin{array}{c|c} \boldsymbol{C}_R & \mathbf{0} \\ \boldsymbol{C}_R \boldsymbol{A}_R & \vdots \\ & \mathbf{0} \\ \vdots & \boldsymbol{C}_S \boldsymbol{E}_S^{\mu-1} \\ & \vdots \\ \boldsymbol{C}_R \boldsymbol{A}_C^{i-1} & \boldsymbol{C}_S \end{array} \right),$$

which is very reminiscent of the Vandermonde structure we encounter in the null space of the Sylvester matrix.

B Algebraic Geometry

An important aim of this thesis is to develop a framework to solve systems of polynomial equations from a linear algebra point of view. Therefore, throughout the thesis we aim to keep the references to the terminology of algebraic geometry minimal.

In the current chapter the basics of polynomial algebra (*i.e.,* algebraic geometry) are presented, including the representation of a polynomial as its coefficient vector, the notions of ideals and varieties, and the definition of the dimension of the solution set of a system of polynomial equations. We have borrowed most of this chapter from Cox et al. (2007, 2005), which provide an excellent and accessible introduction to the field of algebraic geometry.

B.1 Monomials and Polynomials

We will now discuss the main objects of polynomial algebra, namely monomials and polynomials.

Definition B.1. A *monomial* in the variables $x_1, \ldots, x_n$ is a product of the form

$$x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where the exponents $\alpha_1, \ldots, \alpha_n$ are nonnegative integers.

The notation for monomials will usually be simplified as follows: let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ be an n-tuple of non-negative integers. Then we write

$$\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n}.$$

Notice that, when $\boldsymbol{\alpha} = (0, \ldots, 0)$, we have $\boldsymbol{x}^{\boldsymbol{\alpha}} = 1$.

Definition B.2 (Total Degree)**.** The *total degree* of the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ is defined as

$$\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) := |\boldsymbol{\alpha}| := \sum_{i=1}^{n} \alpha_i.$$

Often, we will need to count the number of monomials. The following formulas express the number of monomials, either of total degree *equal to d* or of total degree *less than or equal to d* as binomial coefficients.

Lemma B.3 (Number of monomials)**.** The number of monomials of total degree equal to d in n variables $x_i, \dots, x_n$ is given by

$$\binom{n+d-1}{d} = \binom{n+d-1}{n-1} = \frac{(n+d-1)!}{(n-1)!\,d!}. \tag{B.1}$$

The number of monomials of total degree less than or equal to d in n variables $x_i, \dots, x_n$ is given by

$$\binom{n+d}{d} = \frac{(n+d)!}{n!\,d!}. \tag{B.2}$$

An intuitive way for to understand these numbers is to consider the following related counting problem. Assume that there are n boxes (one for each variable) over which one has to distribute d marbles (degrees). Then the number of possible configurations can be counted as follows: the number of permutations of the $n-1$ separators between the boxes plus the d marbles, divided by the permutations of the separators and the permutations of the marbles. This yields (B.1). Summation over d results in (B.2).

Remark B.4. It can easily be shown that the number of monomials in a fixed number of variables increases polynomially with the degree d. From (B.2) we find, for n fixed,

$$\begin{aligned}\binom{n+d}{d} &= \frac{(n+d)!}{n!\,d!} \\ &= \frac{n^d + O(n^{d-1})}{d!},\end{aligned}$$

where $O(n^{d-1})$ represents the terms in the expansion of degree less than d.

We are now ready to define multivariate polynomials, the central object of study in the current thesis.

Definition B.5. A *polynomial* f in the variables $x_1, \dots, x_n$ is a finite linear combination of monomials. A polynomial f can be written in the form

$$f = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}},$$

where the sum is taken over a finite number of n-tuples $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$. We call $c_{\boldsymbol{\alpha}}$ the *coefficient* of the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$. If $c_{\boldsymbol{\alpha}} \neq 0$, then we call $c_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}}$ a *term* of f. The *total degree* of the polynomial f, denoted $\deg(f)$, is the maximum $\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) = \sum_{i=1}^{n} \alpha_i$ of which the corresponding coefficient $c_{\boldsymbol{\alpha}}$ is nonzero.

Notice that, in this definition, we have not explicitly specified the set out of which the coefficients are taken. In most cases, we will use real numbers as the coefficients of the polynomials. However, without loss of generality, it can be assumed that we are working over the complex numbers, for mathematical convenience. When the coefficients are real numbers, the roots are either real or complex conjugated pairs.

Example B.6. The polynomial $f = 2x_1^3x_2 + 4x_1x_2^2x_3 - 5x_3$ has three terms, and total degree four. Notice that there are two terms of maximal total degree, which is something that cannot happen for polynomials of one variable.

It will often turn out to be necessary to carefully order monomials, for which we have chosen to use the degree negative lexicographic ordering.[1]

Definition B.7 (Degree Negative Lexicographic Order)**.** Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^n$ be monomial exponent vectors. Then two monomials represented by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are ordered by the degree negative lexicographic order as $\boldsymbol{\alpha} <_{\text{dnlex}} \boldsymbol{\beta}$ (simplified as $\boldsymbol{\alpha} < \boldsymbol{\beta}$), if

- $|\boldsymbol{\alpha}| < |\boldsymbol{\beta}|$, or
- $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}|$ and in the vector difference $\boldsymbol{\beta} - \boldsymbol{\alpha} \in \mathbb{Z}^n$, the left-most non-zero entry is negative.

Example B.8. The monomials of maximal degree three in two variables x_1 and x_2 are ordered by the degree negative lexicographic order as

$$1 < x_1 < x_2 < x_1^2 < x_1x_2 < x_2^2 < x_1^3 < x_1^2x_2 < x_1x_2^2 < x_2^3.$$

B.2 Vector Representation of a Polynomial

In this thesis we study linear algebra methods for solving polynomial equations. We will therefore use the well-known property that the polynomials of n variables of (total) degree d form a vector space. A polynomial can hence be represented as a vector. We consider therefore a row vector containing the coefficients (together with zeros) multiplied with a column vector containing all monomials of the n unknowns of degree at most d. The monomials in the monomial vector are ordered by the degree negative lexicographic monomial ordering scheme.

Example B.9. The polynomial $f = 5 - 8x_1 + 4x_2^2$ can be written as the inner product

$$f = \left(\begin{array}{c|cc|ccc} 5 & -8 & 0 & 0 & 0 & 4 \end{array} \right) \left(\begin{array}{c|cc|ccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array} \right)^T,$$

where the bars | separate the blocks of equal total degree.

[1] Most of the results in this thesis immediately hold for any graded monomial ordering.

Lemma B.10. The dimension of the vector space of polynomials of degree d in n variables is given by

$$\binom{n+d}{n}.$$

B.3 Ideals and Varieties

Ideals and varieties play central roles in the questions studied in this thesis. An ideal is generated by a system of multivariate polynomial equations and describes a set of points in the affine space that satisfy all the constituting equations. This solution set is called a variety. In this text we will mainly restrict our attention to the case of zero-dimensional varieties, *i.e.,* the solution set of a system of polynomial equations consisting of a finite number of points.

Let us start by reviewing the definition of the polynomial ring.

Definition B.11. The set of all polynomials in the variables $x_1, x_2, \ldots, x_n$ with coefficients in $\mathbb{C}$ is called a *polynomial ring*, and denoted by $\mathbb{C}[x_1, \ldots, x_n]$.

The next crucial concept is the affine space over the field of coefficients.

Definition B.12. Given the field $\mathbb{C}$ and a positive integer n, the n-dimensional *affine space* over $\mathbb{C}$ is defined as the set

$$\mathbb{C}^n = \{(a_1, \ldots, a_n) : a_1, \ldots, a_n \in \mathbb{C}\}.$$

The core problem we address in this manuscript, is to find the solutions of systems of polynomial equations.

Definition B.13. In terms of algebraic geometry, the basic geometric object concerned with a system of polynomial equations is an *affine variety*: $\mathbf{V}(f_1, \ldots, f_s) \subset k^n$ is the set of all solutions of the system of equations $f_1(x_1, \ldots, x_n) = \ldots = f_s(x_1, \ldots, x_n) = 0$.

The next algebraic object playing a fundamental role in algebraic geometry is the ideal.

Definition B.14. A subset $I \subset \mathbb{C}[x_1, \ldots, x_n]$ is an *ideal* if it satisfies:

1. $0 \in I$.
2. If $f, g \in I$, then $f + g \in I$.
3. If $f \in I$ and $h \in \mathbb{C}[x_1, \ldots, x_n]$, then $hf \in I$.

Naturally, a system of polynomial equations also defines an ideal.

Definition B.15. Let $f_1, \ldots, f_s$ be polynomials in $\mathbb{C}[x_1, \ldots, x_n]$. Then we set

$$\langle f_1, \ldots f_s \rangle = \left\{ \sum_{i=1}^{s} h_i f_i : h_1, \ldots, h_s \in \mathbb{C}[x_1, \ldots, x_n] \right\}.$$

A crucial observation is that $\langle f_1, \ldots f_s \rangle$ is an ideal. We call $\langle f_1, \ldots f_s \rangle$ the *ideal generated by* $f_1, \ldots f_s$.

It turns out that there is an interesting correspondence between ideals and varieties, linking algebra and geometry.

Definition B.16. Let $V \subset \mathbb{C}^n$ be an affine variety. Then we set

$$\mathbf{I}(V) = \{ f \in \mathbb{C}[x_1, \ldots, x_n] : f(a_1, \ldots, a_n) = 0 \text{ for all } (a_1, \ldots, a_n) \in V \}.$$

A crucial observation is that $\mathbf{I}(V)$ is an ideal.

Lemma B.17. If $V \subset \mathbb{C}^n$ is an affine variety, then $\mathbf{I}(V) \subset \mathbb{C}[x_1, \ldots, x_n]$ is an ideal. We will call $\mathbf{I}(V)$ the *ideal of* V.

B.4 Projective Ideals and Varieties

One of the interesting properties of projective geometry is that points at infinity are incorporated as regular points for which the homogenization variable x_0 takes the value zero. We will see that our matrix formulations are implicitly always describing solutions in the projective space.

The projective space is defined by the equivalence relation $\sim$ on the $n+1$-dimensional space by setting

$$(x_0', \ldots, x_n') \sim (x_0, \ldots, x_n),$$

such that $(x_0', \ldots, x_n') = \lambda(x_0, \ldots, x_n)$ with $\lambda \neq 0$.

The projective space is hence defined as follows.

Definition B.18. The *n-dimensional projective space over* $\mathbb{C}$ is defined as the set of equivalence classes of $\sim$ on $\mathbb{C}^{k+1} - \{0\}$. Each nonzero $n+1$-tuple $(x_0, \ldots, x_n)$ defines a point in the projective space, and we call $(x_0, \ldots, x_n)$ its *projective coordinates*.

For many purposes, the projective space can be treated similarly as the affine space, with the difference being an additional dimension. However, it turns out that care must be taken when the notions of ideals and varieties are considered for the projective case. In the case one is working in the projective space, *homogeneous* polynomials, *i.e.,* polynomials in which all terms are of equal total degree, must be considered. The following definition states how the so-called homogenization of a polynomial can be obtained.

Definition B.19 (Homogenization and dehomogenization)**.** The homogenization of an equation f, denoted f^h, is computed using the formula

$$f^h = x_0^d \cdot f\left(x_1/x_0, \dots, x_n/x_0\right).$$

Dehomogenizing f^h yields f, or formally

$$f^h(1, x_1, \dots, x_n) = f(x_1, \dots, x_n).$$

The variable x_0 is sometimes called the *homogenization variable*.

Now, the notions of ideal and variety can be naturally generalized. For instance, we have

Definition B.20. Let $f_1, \dots, f_s$ homogeneous polynomials. Then we set

$$\mathbf{V}(f_1, \dots, f_s) = \left\{(a_0, \dots, a_n) : f_i(a_0, \dots, a_n) = 0 \text{ for all } 1 \leq i \leq s\right\}.$$

We call $\mathbf{V}(f_1, \dots, f_s)$ the *projective variety* defined by $f_1, \dots, f_s$.

B.5 Dimension of a Variety

There are several definitions of the dimension of a variety in algebraic geometry. Some of the definitions are of geometric nature, whereas others are purely algebraic. For this thesis, two definitions are of interest, namely the one using the Hilbert function and one based on the more intuitive algebraic and geometric notions of independence.

B.5.1 Intuitive Definition

Throughout this manuscript, an intuitive notion of dimension will suffice in most cases. We illustrate the idea by the case where a variety V is defined by a single polynomial. We have that the dimension of the variety, defined by a homogeneous polynomial f, in the projective space is $\dim \mathbf{V}(f) = n - 1$. Indeed, this corresponds to our intuitive geometric notion of an equation imposing a constraint on the free variables: a single equation reduces the degrees of freedom by one.

Under mild conditions, this intuitive notion can be generalized to the following proposition.

Proposition B.21. Consider the homogeneous polynomials $f_1, \dots, f_s$. We have that

$$\dim \mathbf{V}(f_1, \dots, f_s) = n - s.$$

Said in words, in generic conditions, every equation makes the dimensionality of the space drop with one — each equation diminishes the available degrees of freedoms by one.

More precisely, we have the following theorem.

Theorem B.22 (Cox et al. (2007))**.** Let $V := \mathbf{V}(f_1, \ldots, f_s) \in \mathbb{C}^n$ be a variety and suppose that $\boldsymbol{x}^\star \in V$ is a point on V where the Jacobian matrix $\boldsymbol{J}_f(\boldsymbol{x}^\star)$ has rank s (see (1.2)). Then $\boldsymbol{x}^\star$ is a nonsingular point of V and lies on a component of V of dimension $n - s$.

The theorem supports the intuitive notion that the dimension of a variety should drop by one for every equation defining the variety. Moreover, it specifies the condition under which this happens: the defining equations $f_1, \ldots, f_s$ should be sufficiently independent (expressed by the rank constraint on $\boldsymbol{J}_f$).

B.5.2 Definition Using Hilbert Polynomial

There is another definition of dimension having an algebraic nature and due to Hilbert (1890). Hilbert's notion of dimension arose from the insight that the dimension of a monomial ideal is characterized by the increase of the monomials *not* in the ideal as the total degree increases.

A careful generalization of this notion to *any* ideal leads to the formulation of the (affine) Hilbert function, defined as

$$\begin{aligned} {}^aHF_I(d) &= \dim \mathbb{C}[x_1, \ldots, x_n]_{\leq d} / I_{\leq d}, \\ &= \dim \mathbb{C}[x_1, \ldots, x_n] - \dim I_{\leq d}, \end{aligned}$$

where $\mathbb{C}[x_1, \ldots, x_n]_{\leq d}$ denotes the set of polynomials of total degree $\leq d$ in $\mathbb{C}[x_1, \ldots, x_n]$, and $I_{\leq d} := I \cap \mathbb{C}[x_1, \ldots, x_n]_{\leq d}$.

For a sufficiently large degree $d \geq d_H$, called the index of regularity of I, the affine Hilbert function ${}^aHF_I(d)$ coincides with a polynomial, called the (affine) Hilbert polynomial ${}^aHP_I(d)$.

Theorem B.23 (Dimension of a Variety using Hilbert Polynomial)**.** The dimension of the variety $V := \mathbf{V}(I)$ is defined as the degree of the Hilbert polynomial ${}^aHP_I(d)$, or

$$\dim V = \deg {}^aHP_I.$$

B.6 Gröbner Bases and Buchberger's Algorithm

Knowing + and × is good enough,
understanding their interaction is
ideal.

BRUNO BUCHBERGER

B.6.1 Introduction

Buchberger (1965) developed the algorithm that sparked the beginning of computer algebra. The so-called Gröbner basis algorithm computes a nicely behaved basis for a given set of polynomial equations. Loosely speaking, the procedure can be understood as the polynomial generalization of the Gaussian elimination algorithm and Euclidean GCD algorithm to multivariate polynomials.

In essence, Buchberger's algorithm proceeds by manipulating the given set of equations with the objective of eliminating certain terms, while at all times the 'new' set of equations is algebraically equivalent to the given equations, *i.e.,* defining the same ideal and hence describing the same solution set.

A typical way to do this is to aim at finding a triangular structure: one or more equations would be univariate, then some equations would involve two or a few variables, until the most complicated equation involves (almost) all variables. This requires that one defines an ordering on the monomials to decide in what order the terms should be eliminated.

The triangular structure greatly simplifies the task of solving the system. For instance, if a lexicographic ordering is used, one of the polynomials in the result will be univariate. After the univariate equation can be solved, the solutions can be substituted in the next equation which has two unknowns, from which the second unknowns can again be determined, *etc.* By iterating in this way, all unknowns are ultimately determined. Not only lexicographic term orderings are considered; the properties of a Gröbner basis with different term orderings (*e.g.,* total degree orderings (Cox et al., 2007)) are in several circumstances even more desirable.

Gröbner bases would become the backbone of nearly all computational algebraic geometry methods, and is until today the most dominant tool in computer algebra. We have deliberately avoided the precise description of the Gröbner basis algorithm until this point, because we aim to show how a numerical linear algebra approach can be used as an alternative for finding the zeros of a system of polynomial equations. For the sake of completeness, and in order to fully describe the Stetter-Möller matrix method, which is closely related to our matrix method, the definition of a Gröbner basis is presented now.

Much of the material in this section is taken from Cox et al. (2007).

B.6.2 General Definitions and Multivariate Division

An important ingredient in Buchberger's algorithm is a term ordering. For example the degree negative lexicographic ordering we have defined in Definition 5.1 can be used. In the literature, a number of different orderings is described, but their enumeration is not of any relevance for the current manuscript. Let us briefly recall two of the well-known ones: lexicographic ordering (analogous to the ordering of words in a dictionary) and graded lexicographic ordering.

Definition B.24. *(Lexicographic Ordering)* Let $\boldsymbol{x}^{\boldsymbol{\alpha}}$ and $\boldsymbol{x}^{\boldsymbol{\beta}}$ be monomials in $\mathbb{C}[x_1,\ldots,x_n]$. We say $\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\text{lex}} \boldsymbol{x}^{\boldsymbol{\beta}}$ if in the difference $\boldsymbol{\alpha} - \boldsymbol{\beta} \in \mathbb{Z}^n$, the leftmost nonzero entry is positive.

Definition B.25. *(Graded Lexicographic Ordering)* Let $\boldsymbol{x}^{\boldsymbol{\alpha}}$ and $\boldsymbol{x}^{\boldsymbol{\beta}}$ be monomials in $\mathbb{C}[x_1,\ldots,x_n]$. We say $\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\text{grlex}} \boldsymbol{x}^{\boldsymbol{\beta}}$ if $\sum_{i=1}^{n} \alpha_i > \sum_{i=1}^{n} \beta_i$, or if $\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i$, and $\boldsymbol{x}^{\boldsymbol{\alpha}} >_{\text{lex}} \boldsymbol{x}^{\boldsymbol{\beta}}$.

Let us assume that a monomial ordering is chosen, expressed by >, and let us now consider the terms appearing in a given polynomial $f = \sum_{\alpha} c_{\alpha} x^{\alpha}$ and introduce some definitions.

Definition B.26 (Leading Term, Leading Coefficient, Leading Monomial)**.** The *leading term* of f (w.r.t. >) is the product $c_{\alpha}\boldsymbol{x}^{\boldsymbol{\alpha}}$ where $\boldsymbol{x}^{\boldsymbol{\alpha}}$ is the largest monomial appearing in f in the ordering >. The notation $LT(f)$ will be used for the leading term. Furthermore, if $LT(f) = c\boldsymbol{x}^{\boldsymbol{\alpha}}$, then $LC(f) = c$ is the *leading coefficient* of f and $LM(f) = \boldsymbol{x}^{\boldsymbol{\alpha}}$ is the *leading monomial.*

A *multivariate division algorithm* can now be devised, for which an algorithmic implementation is presented below. Let us first set up some basics.

Definition B.27 (Multivariate Division)**.** Fix any monomial order >, and let $F = f_1,\ldots,f_s$ be an ordered set of polynomials. Then any polynomial $f \in \mathbb{C}[x_1,\ldots,x_n]$ can be written as

$$f = a_1 f_1 + a_2 f_2 + \ldots + a_s f_s + r, \tag{B.3}$$

where $a_i, r \in \mathbb{C}[x_1,\ldots,x_n]$, for each i, $a_i f_i = 0$ or $LT(f) \geq LT(a_i f_i)$, and either $r = 0$, or r is a linear combination of monomials, none of which is divisible by any of $LT(f_1),\ldots,LT(f_s)$. We call r a *remainder* of f on division by F.

Algorithm 8 provides a procedure to perform the multivariate division.

Algorithm 8. *(Multivariate Division Algorithm)*

```
input:  $f_1, \ldots, f_s, f$
output: $a_1, \ldots, a_s, r$

  1) $a_1 := 0; \ldots; a_s := 0; r := 0$
  2) $p := f$
  3) while $p \neq 0$, do
       a) $i := 1$
       b) divisionoccurred := false
       c) while $i \leq s$ and divisionoccurred = false, do
            i. if $LT(f_i)$ divides $LT(p)$, then
                 A. $a_i := a_i + LT(p)/LT(f_i)$
                 B. $p := p - (LT(p)/LT(f_i))f_i$
                 C. divisionoccurred := true
            ii. else
                 A. $i := i + 1$
       d) if divisionoccurred = false, then
            i. $r := r + LT(p)$
            ii. $p := p - LT(p)$
```

B.6.3 From S-Polynomials to Buchberger's Algorithm

Another central ingredient in Buchberger's algorithm are the so-called *S-polynomials* (from subtraction polynomials, or syzygy polynomials). S-Polynomials are used to eliminate leading terms from a system of equations by considering two polynomials p and q from the system, and consequently computing a *least common multiple*, leading to the vanishing of the leading terms.

The multivariate division algorithm is used to reduce the result from this operation to a normal form with respect to a set of polynomial equations (this normal form is the remainder r in the multivariate division algorithm).

Let us consider the following definition.

Definition B.28 (S-Polynomial)**.** Let $f, g \in \mathbb{C}[x_1, \ldots, x_n]$ be nonzero polynomials. Fix a monomial order $>$ and let

$$LT(f) = c\boldsymbol{x}^{\boldsymbol{\alpha}} \quad \text{and} \quad LT(g) = d\boldsymbol{x}^{\boldsymbol{\beta}},$$

where $c, d \in k$. Let x^γ be the least common multiple of x^α and x^β. The *S-polynomial* of f and g, denoted $S(f,g)$, is the polynomial

$$S(f,g) := \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g.$$

By definition, $S(f,g) \in \langle f, g \rangle$.

Example B.29. Consider $f = x^3y - 2x^2y^2 + x$ and $g = 3x^4 - y$ in $\mathbb{Q}[x,y]$, and using $>_{\text{lex}}$, we have $x^\gamma = x^4y$, and

$$S(f,g) = xf - (y/3)g = -2x^3y^2 + x^2 + y^2/3.$$

If we continue with the equations from the previous example, and now take the remainder on division by $F = (f,g)$, denoted $\overline{S(f,g)}^F$, we can uncover new leading terms of elements in $\langle f, g \rangle$. Note that this step requires the multivariate division algorithm as described above, and essentially reduces the result from the S-polynomial with respect to the system of equations we are dealing with. In this case, we find that the remainder is

$$\overline{S(f,g)}^F = -4x^2y^3 + x^2 + 2xy + y^2/3,$$

and $LT\left(\overline{S(f,g)}^F\right) = -4x^2y^3$ is divisible by neither $LT(f)$ nor $LT(g)$. Buchberger's algorithm consists in identifying such interfering polynomials in the system, computing the normal form (*i.e.*, the remainder in the division algorithm), adding the result to the set of equations, and repeating this procedure until all reductions yield zero. The following theorem expresses a criterion by Buchberger that is essential in this sense.

Theorem B.30. A finite set $G = g_1, \ldots, g_t$ is a Gröbner basis of $I = \langle g_1, \ldots, g_t \rangle$ iff $\overline{S(g_i, g_j)}^G = 0$ for all pairs $i \neq j$.

This exposition leads to a rudimentary version of Buchberger's algorithm, presented in Algorithm 9.

Algorithm 9. *(Buchberger's Algorithm)*

```
input:  Set of Polynomial Equations
        $F = \{f_1, \ldots, f_s\}$
output: Reduced Gröbner Basis
        $G = \{g_1, \ldots, g_t\}$ for $I =< F >$

  1) $G := F$
  2) repeat
      a) $H := G$;
```

```
    b) for each pair $p \neq q$ in $H$, do
        i. $h := \overline{S(p,q)}^{H}$
        ii. if $h \neq 0$, do $G := G \cup \{h\}$
until $G = H$
```

Definition B.31 (Reduced Gröbner Basis, Monic Gröbner Basis). A *reduced Gröbner basis* for an ideal $I \subset \mathbb{C}[x_1, \ldots, x_n]$ is a Gröbner basis G for I such that for all distinct $p, q \in G$, no monomial appearing in p is a multiple of $LT(q)$. A *monic Gröbner basis* is a reduced Gröbner basis in which the leading coefficient of every polynomial is 1, or $\varnothing$ if $I = \langle 0 \rangle$.

Although Buchberger's rudimentary algorithm is constructive and finishes in a finite number of steps (albeit its worst-case complexity is $d_\circ^{2^{O(n)}}$, where n is the number of variables, and $d_\circ$ is the max degree of the equations, *i.e.*, $d_\circ := \max \deg(f_i)$), it requires exact arithmetic, and turns out to be impractical for even medium-sized problems. During the last decades, a series of optimizations has been introduced, of which the work by Faugère (1999, 2002) are currently the most competitive approaches (also among other classes of solution methods).

B.7 Stetter's Eigendecomposition Approach

In the current section we will illustrate the Stetter approach (sometimes called the Stetter-Möller approach) for solving a system of polynomial equations as an eigenvalue problem. The approach works by computing an eigenvalue decomposition of a matrix representing multiplication in the quotient space $\mathbb{C}[x_1, \ldots, x_n]/I$ with $I := \langle f_1, \ldots, f_s \rangle$.

We assume that ideal $I := \langle f_1, \ldots, f_s \rangle$ is radical, *i.e.*, $\sqrt{I} = I$ with $\sqrt{I} := \{g \in \mathbb{C}[x_1, \ldots, x_n] : g^\mu \in I \text{ for some } \mu \geq 1\}$, and describes a zero-dimensional variety. Let G be a Gröbner basis of I. The quotient space $\mathbb{C}[x_1, \ldots, x_n]/I$ is an m-dimensional vector space and has as a monomial basis the set of m monomials that do *not* lie in the ideal spanned by the leading terms of G, which we denote by B.

Any polynomial $f \in \mathbb{C}[x_1, \ldots, x_n]$ can now be reduced modulo $f_1, \ldots, f_s$ to a linear combination of the monomials in B. Moreover, multiplication in $\mathbb{C}[x_1, \ldots, x_n]$ can be represented by a multiplication operator, defined as

$$\begin{aligned} \mathcal{A}_{x_i} : \mathbb{C}[x_1, \ldots, x_n]/I &\to \mathbb{C}[x_1, \ldots, x_n]/I \\ g &\mapsto x_i \cdot g. \end{aligned}$$

This operator defines an $m \times m$ multiplication matrix $\boldsymbol{A}_{x_i}$. The matrices $\boldsymbol{A}_{x_i}$ and $\boldsymbol{A}_{x_j}$ for $i, j \in \{1, \ldots, n\}$ commute since $x_i \cdot x_j = x_j \cdot x_i$. As a result, the matrices $\boldsymbol{A}_{x_i}$ have common eigenspaces. It can moreover be shown that the eigenvalues $\lambda_i, i = 1, \ldots, n$ associated to a common eigenvector v are the i-th component of the points on the variety $\mathbf{V}(I)$. From the eigenvectors, the mutual matching among the components x_i can be retrieved.

Summarizing, the approach proceeds as follows:

1. A basis for the quotient space $\mathbb{C}[x_1, \ldots, x_n]/I$ is obtained by computing a Gröbner basis G. The normal set B of the Gröbner basis G serves as a basis for the quotient space and are placed in the Stetter (eigen)vector. The normal set B is defined as the set of monomials that do not lie in the ideal spanned by the leading terms of G.

2. For any given multiplication polynomial $g(x_1, \ldots, x_n)$, the shifts of the normal set monomials with g are reduced to an expression in terms of the normal set elements. This is done using a normal form algorithm. The normal form is the remainder r in the polynomial division procedure, expressed in terms of the monomials that are not in the ideal of G. In this way, the rows of a so-called multiplication matrix $\boldsymbol{A}_g$ are constructed.

3. The eigenvalues of the multiplication matrix $\boldsymbol{A}_g$ provide the evaluations of the roots (*i.e.,* the points of $\mathbf{V}(I)$, evaluated in the shift polynomial g. Properly rescaling the eigenvectors reveals the monomials of the normal set and hence the mutual matching between the solution components.

Example B.32. Let us show this approach on a small example. We revisit the equations of Example 6.13. The equations are

$$\begin{array}{rcl} f_1(x_1, x_2, x_3) & = & x_1 x_2 - 3 = 0 \\ f_2(x_1, x_2, x_3) & = & x_1^2 - x_3^2 + x_1 x_3 - 5 = 0 \\ f_3(x_1, x_2, x_3) & = & x_3^3 - 2x_1 x_2 + 7 = 0, \end{array}$$

and a Gröbner basis G is computed using the degree negative lexicographic ordering. We find

$$G = \left\{ \begin{array}{rcl} x_1 x_2 - 3 & = & 0, \\ 14x_2^2 - 25 - 5x_1 x_3 + 2x_3 - 5x_2 + x_1 & = & 0, \\ 5x_3 x_2 + 15 - 6x_1 x_3 - 3x_1^2 - x_2 & = & 0, \\ x_3^2 + 5 - x_1 x_3 - x_1^2 & = & 0, \\ 5x_1^3 - 25 + 4x_1 x_3 + 2x_1^2 - 25x_3 + 84x_2 - 75x_1 & = & 0, \\ 5x_3 x_1^2 + 15 - 2x_1 x_3 - x_1^2 - 42x_2 + 25x_1 & = & 0. \end{array} \right.$$

The leading terms of G are $\left\{x_1 x_2, x_2^2, x_2 x_3, x_3^2, x_1^3, x_1^3 x_3\right\}$, so the normal set is

$$B = \left\{1, x_1, x_2, x_3, x_1^2, x_1 x_3\right\}.$$

Next, we choose $g(x_1, x_2, x_3) = x_1 + 2x_2 + 3x_3$ as the shift polynomial as in Example 6.13 and we have the multiplication matrix $\boldsymbol{A}_g$ as

$$\boldsymbol{A}_g = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 3 \\ -17/7 & -1/7 & 46/35 & -2/7 & 9/5 & 151/35 \\ -21 & 0 & 2/5 & 0 & 21/5 & 32/5 \\ -4 & 6 & 42/5 & 5 & 1/5 & 2/5 \\ 3 & 10 & -84/5 & 21 & -2/5 & -4/5 \end{pmatrix},$$

which is obtained by computing for each element of B its multiplication with g and reducing the result with respect to the Gröbner basis G; the remainder is linear in the elements of B and composes a row of $\boldsymbol{A}_g$.

We have then

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 3 \\ -17/7 & -1/7 & 46/35 & -2/7 & 9/5 & 151/35 \\ -21 & 0 & 2/5 & 0 & 21/5 & 32/5 \\ -4 & 6 & 42/5 & 5 & 1/5 & 2/5 \\ 3 & 10 & -84/5 & 21 & -2/5 & -4/5 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_1^2 \\ x_1x_3 \end{pmatrix} = (x_1 + 2x_2 + 3x_3) \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_1^2 \\ x_1x_3 \end{pmatrix}.$$

The eigenvectors of $\boldsymbol{A}_g$ are rescaled such that the first entry equals one. From this we read off the solutions as

x_1	x_2	x_3
$1.857 \mp 0.176i$	$1.600 \pm 0.151i$	$0.500 \pm 0.866i$
-2.000	-1.500	-1.000
$-2.357 \pm 0.689i$	$-1.172 \mp 0.343i$	$0.500 \mp 0.866i$
3.000	1.000	-1.000

C Polynomial System Solving: Historical Notes

C.1 Introduction

The problem of finding the roots of a polynomial, or a system of multivariate polynomials, is one of the oldest questions in mathematics and an essential task in scientific computing. It arises at many problems in science and engineering, and has a very long and rich history that be traced back to the Sumerians, ancient Egypt and Babylon, where the problem originated in mensuration and surveying problems. The problem has defined the very course of mathematical development for thousands of years.

In the current chapter, some historical and bibliographic notes and anecdotes regarding the development of algebra, geometry, and algebraic geometry have been collected. Emphasis is put on the facts concerning the task of solving systems of polynomial equations and linear algebra. This chapter is heavily based on Pan (1997); Smith (1951, 1953); Stewart (1993), which provide a most intriguing collection of historical facts about the history of mathematics in general, and specifically the history of the problems addressed in this thesis.

C.2 Pre-history

The task of solving a polynomial equation has historically motivated the development of several fundamental mathematical concepts. The interaction between algebra and geometry has been central throughout the development of mathematics: often, algebraic problems were solved by geometric methods.

The **Sumerians** (3rd millennium BCE) knew in some way the problem of finding the solutions of a polynomial equation. The problem occurs also in the ancient mathematics developed in **Egypt** and **Babylon**. The questions were stated in words (*i.e.,* rhetorical), although traces of symbolic algebra have been found.

The Egyptians had methods for solving linear equations around 1800 BCE and were able to solve systems of two equations in two unknowns around 300 BCE. The mathematics of Babylon (1800-1600 BCE) was already more advanced than that of Egypt: they considered problems involving more than two unknowns and equations of higher degrees. Solution procedures were presented through the use of examples; reasons and explanation were omitted.

The word 'geometry' is derived from the Greek words for 'earth' and 'measurement' and was the central feature of the mathematics of the **Greeks** and often arising from mensuration problems of simple geometric objects or the study of symmetry. The Greek mathematicians viewed problems and their proposed solutions from a geometrical viewpoint, without attempting to demonstrate the reasoning behind them.

Euclid of Alexandria (*fl.* 300 BCE) is often considered as the father of geometry. His 'Elements' would serve as the main textbook for geometry from the time of its publication until the late 19th and early 20th century. Euclid devised a geometrical method for solving a quadratic equation. Several typical geometric problems were tackled by finding the intersections of algebraic curves. He is also known for Euclid's algorithm, a procedure to compute the greatest common divisor of two numbers, one of the oldest algorithms still in use today.

The later Greek mathematician **Diophantus of Alexandria** (250 CE) turned away from the purely geometrical viewpoint: In 'Arithmetica', he gives a treatment of indeterminate equations, usually in two or more equations in several variables that have an infinite number of rational solutions. Diophantus was the first to introduce symbols for the unknowns, together with other algebraic symbols. General methods were lacking: each of the 189 problems in Arithmetica are solved by different methods.

The problems that were studied in **Hindu algebra** (800 BCE) were mainly motivated by astronomy and astrology and were in their later development significantly influenced by Greek mathematics. Only around 600 CE, their base 10 positional numeral system had become standard. From then on, the number zero was treated as a true number, and operations involving this new number were studied.

In the 7th and 8th centuries CE, the **Arabs** conquered the land from India, across northern Africa, to Spain. In the following centuries (until the 14th century CE), many mathematical and scientific developments were made.

From the viewpoint of algebraic geometry, the Arab mathematicians were able to solve — by purely algebraic means — certain cubic equations, and were then able to interpret the results geometrically.

An important contribution from the same era was the word 'algebra' that is derived from the title of a text book in the subject, 'Hisab al-jabr w'al muqabala', written about 830 CE by the Persian astronomer-mathematician **Mohammed ibn-Musa al-Khowarizmi**. The word 'algorithm' is a corruption of his name. Subsequently, the Persian mathematician **Omar Khayyám** (born 1048 CE) discovered a general method of solving cubic equations by intersecting a parabola with a circle.

In the following centuries, a lot of interesting developments in mathematics were made in **China**. **Zhu Shijie** wrote his second book, 'Jade Mirror of the Four Unknowns' in 1303 CE. The first four solutions of the 288 problems illustrate his method of the four unknowns. He shows how to convert a problem stated verbally into a system of polynomial equations (up to degree fourteen), and how to reduce the system to a single polynomial equation in one unknown, which he solves by **Qin Jiushao**'s method published in 'Mathematical Treatise in Nine Sections' in 1247 CE, making use of a diagram, nowadays known as the Pascal triangle.

Preceding the work of **Carl Friedrich Gauss** by 500 years, Zhu Shijie showed how to solve systems of linear equations be reducing the matrix of their coefficients to a diagonal form. Rather amazingly, many of the methods described by Zhu pre-date those by **Blaise Pascal** (1623-1662 CE), **William Horner** (1786-1837 CE), and modern matrix methods by centuries.

C.3 Renaissance

Techniques in which geometrical constructions are applied to algebraic problems were adopted by a number of **Italian** Renaissance mathematicians such as **Gerolamo Cardano**, **Scipione del Ferro** and **Niccolò Fontana Tartaglia** on their studies of the cubic equation. The development of imaginary numbers stemmed from the work of **Rafael Bombelli** around the same time period. The geometrical approach to construction problems, rather than the algebraic one, was favored by most 16th and 17th century mathematicians.

The birth of analytic geometry was possible due to three important developments: a coordinate system, a one-to-one correspondence between algebra and geometry, and a graphical representation of an algebraic expression. Although coordinate systems were known to Greek, Arab, Persian and Hindu, it were the **French** mathematicians **Franciscus Vieta** and later **René Descartes** and **Pierre de Fermat** who revolutionized the conventional way of thinking about construction problems through the introduction of coordinate geometry.

During the same period, **Blaise Pascal** and **Gérard Desargues** approached geometry from a different perspective, developing the synthetic notions of projective geometry. Ultimately, the analytic geometry of **Descartes** and **Fermat** would supply the 18th century mathematicians with quantitative tools needed to study physical problems using the new calculus of **Isaac Newton** and **Gottfried Wilhelm Leibniz**. By the end of the 18th century, most of the algebraic character of coordinate geometry was subsumed by the calculus of infinitesimals of **Joseph-Louis Lagrange** and **Leonhard Euler**. Later, the renaissance of pure geometry, beginning in the 19th century and characterized by the projective geometry of **Jean-Victor Poncelet**, would ultimately lead to the non-Euclidean hypotheses of **Nikolai Lobachevsky**, **János Bolyai**, and **Bernhard Riemann**.

The Italian algebraists of the 16th century assumed that every rational integral equation has a root. The first writer to assert positively that every polynomial equation of the *n*th degree has *n* roots and no more seems to have been **Peter Roth**. **Descartes** (1637) more clearly expressed the law, but distinguished between real and imaginary roots and between positive and negative real roots in making the total number. After these early steps the statement was repeated in one form or another by various later writers, but the first rigorous demonstration is due to **Gauss** (1799).

Around 1770, **Lagrange** started to study resolvents to unify the many different tricks to solve polynomial equations. The work was a precursor to **Galois** theory. Lagrange failed to develop methods for solving equations of degree five or higher; however, he could not prove that this was impossible. Later on, this was indeed proved by **Paolo Ruffini** (1799) and **Niels Henrik Abel** (1823). Modern proofs use Galois theory (the first proof using Galois theory dates back to 1846).

C.4 19th Century

During the 19th century, the branch of mathematics that was concerned with solving polynomial equations was elimination theory. The **Sylvester** matrix construction beautifully shows the intimate link between polynomials, matrices and resultants, an important tool in elimination theory that expresses the existence of common solutions by conditions on the coefficients. This insight would ultimately lead to the formulation of eigenproblems. A number of methods exists for constructing resultants matrices, which are matrices whose determinant is the resultant. Important contributions in this field are due to **Leopold Kronecker**, **Étienne Bézout**, **David Hilbert**, **James Joseph Sylvester** and **Francis Sowerby Macaulay**.[1]

The algebraists of the late 19th century, such as **Sylvester** and **Macaulay**, must have already been aware of the connection between polynomial system solving and the multiplicative structure of its quotient space (and, hence, in our current understanding, matrix eigenvalue problems), albeit in the language of their own time: matrix theory was still premature.

Around 1840, the German mathematician **Hermann Grassmann** began investigating vectors. The American physicist **Josiah Willard Gibbs** developed an algebra of vectors in three-dimensional space and recognized in vector algebra a system of great utility for physicists. The widespread influence of this abstract approach led **George Boole** to write 'The Laws of Thought' (1854), an algebraic treatment of basic logic. Since that time, modern algebra — also called abstract algebra — has continued to develop. Important new results have been discovered, and the subject has found applications in all branches of mathematics and in many of the sciences as well.

The mathematical discipline of abstract algebra resulted out of the work of a bright German mathematician, **David Hilbert**. Hilbert is recognized as one of the most influential mathematicians of the 19th and 20th century. Hilbert's famous basis theorem can be translated into terms of algebraic geometry as follows: 'every algebraic set over a field can be described as the set of common solutions of finitely many polynomial equations'.

By the 19th century, the scope of algebra had expanded to the study of algebraic form and structure and was no longer limited to ordinary systems of numbers. The attention shifted from solving polynomial equations to studying the structure of abstract mathematical systems whose axioms were based on the behavior of mathematical objects that mathematicians encountered when studying polynomial equations. Since matrix theory was still in an early stage of development, there was no immediate emphasis on matrix methods. It should therefore come as no surprise that the equivalence of polynomial system solving and matrix eigenvalue problems was only rediscovered at the end of the 20th century.

C.5 20th Century

A driving force throughout the history and development of mathematics so far is a desire for algorithms and computation. Tools and devices for facilitating computation and manipulating numbers have been around since the prehistory of mathematics, ranging from the abacus (2700-2300 BCE) for addition

[1] A modern and systematic version of the theory of the discriminants has been developed by **Israel Moiseevich Gelfand** and coworkers (Gelfand et al., 1994).

and subtraction operations to **Blaise Pascal**'s 'arithmetic machine' (*ca.* 1642) to add and subtract numbers directly and perform multiplication and division by repetition, and many more.

The 20th century would witness a boom in computation, leading to the digital age in which we are living today. Important steps in the history of modern digital computing can be traced back to the 19th century when **Charles Babbage** conceived his difference engines (*ca.* 1820-1824) which were able to tabulate polynomial functions. His later analytical engines (*ca.* 1834) were programmable and are thus considered as the precursor of modern digital computers, because of which Babbage is still considered 'the father of computing'.[2,3] Only well into the 20th century mechanical calculators took central stage in the the process of 'computing' which was until then a laborious task which was mainly taken up by 'human computers'.

During the 1930's and 1940's, **Alan Turing** and **John von Neumann** made substantial contributions that would ultimately shape the field of digital computing. The seminal development of the semiconductor transistor in 1947 sparked the era of digital computing. Since 1965 **Gordon Moore**'s law prescribes the exponential increase of the number of transistors on an integrated circuit, leading to the exponential rise in processing speed and memory capacity, first leading to mainframe computers (*ca.* 1950-1970) and later to the era of personal computing.

The study of matrix algebra appeared in the mid 1800's in England and was studied by **Sylvester** and **Arthur Cayley**, but was at the time mainly focusing on determinants and elimination theory. During the 20th century, matrix algebra was turning out to be applicable to many subjects outside pure mathematics, such as physics, statistics, quantum mechanics, *etc.* and, on the other hand, the advent of digital computing was at hand. The modern use of matrix algebra, and consequently numerical linear algebra, took its form only around the 1950's. Numerical linear algebra would turn out to become an essential tool for modeling and simulation, and is still one of the most important areas of scientific computing today.

The singular value decomposition (SVD) would (much later) become a central computational tool in numerical linear algebra. SVD theory can be traced back to contributions of **Eugenio Beltrami**, **Sylvester**, **Camille Jordan**, **Erhard Schmidt** and **Hermann Weyl**. It was **Gene Golub** and **William Kahan** who proposed the first practical algorithm for computing the SVD in 1965.

[2] Unfortunately, Babbage's devices were never built during his lifetime — the London Science Museum has built a fully operational difference machine on the basis of Babbage's design only in 1990.

[3] Together with the first programmable machine, it was **Ada Lovelace** (1815-1852) who became the first computer programmer: she came up with a set of instructions intended to be processed by a machine such as Babbage's analytical machine.

Due to historical reasons, by the 1950's, the (natural) links between matrix algebra and polynomial system solving had seemed to be abandoned or forgotten. It was only by the end of the 20th century that these links were rediscovered independently by a number of researchers.

One of the new developments in 'computational' algebraic geometry came from the side of algebraic geometry. **Bruno Buchberger** proposes in his PhD thesis (1965) an algorithm for computing a generating set of ideals, having some desirable properties. He coins such a generating set a 'Gröbner basis' in honor of his thesis advisor **Wolfgang Gröbner**.

Being one of the first computational tools in algebraic geometry, Buchberger's approach would dominate the field of 'computer algebra' for decades, despite some major drawbacks. Although the algorithm is guaranteed to end, the computational complexity is rather poor (both in terms of computing time and memory required), since intermediate computations and results can become very large, even for medium-sized problems. Moreover, the algorithm is conceived in a symbolic setting which makes an implementation in floating point arithmetic extremely cumbersome. Nevertheless, due to the huge amount of research activity which has yielded improvements to the rudimentary algorithm for several decades, Gröbner basis techniques are today one of the most used tools in computer algebra.

In the 1980s, due to the independent research developments of a number of scholars, the natural links between polynomial system solving and eigenvalue problems were rediscovered. **Daniel Lazard** rediscovers in the resultant-based framework the work of **Macaulay** and **Sylvester** and illustrates how a algebraic system of polynomials can be solved using matrix computations. He describes how from a Macaulay coefficient matrix a Gröbner basis can be computed by means of Gaussian elimination. Although the emphasis is not on the link to eigenvalue problems specifically, seeds of the eigenvalue method are in this work.

Later, **Hans J. Stetter** and coworkers propose a method for the determination of all isolated zeros of a system of multivariate polynomial equations. By so-called 'polynomial combination', the system is reduced to a special form which is interpreted as a multiplication table for power products (monomials) modulo the ideal generated by the polynomial equations. The zeros are then computed from an ordinary eigenvalue problem from the matrix of the multiplication table; either as the eigenvalues, or they can be read off from the eigenvectors. After Lazard and Stetter, a series of authors further explores the links between polynomial system solving and eigenvalue computations, such as **Dinesh Manocha**, **Bernard Mourrain**, **Ioannis Z. Emiris**, **Victor Pan**, **Guðbjörn Jónsson** and **Stephen Vavasis**, among others.

At the moment, interesting developments in polynomial optimization problems are taking place. For numerically solving a system, probably the homotopy methods of **Tien-Yien Li** and **Jan Verschelde** are currently the most efficient algorithms. For specific subclasses of problems modern methods using positivity or convex optimization, such as the methods of **Pablo Parrilo** and **Jean-Bernard Lasserre**, greatly outperform the classical computer algebra methods. The 21st century may well become the golden century for polynomial algebra.

Curriculum Vitae

Philippe Dreesen was born on January 31, 1982 in Bree, Belgium. In 2007 he received the M.Sc. degree in Electrical Engineering (Burgerlijk Werktuigkundig-Elektrotechnisch Ingenieur, richting Elektrotechniek, optie Dataverwerking en Automatisatie) from the KU Leuven in Belgium. In October 2007 he started his doctoral studies at KU Leuven/ESAT-STADIUS on the topic of solving systems of polynomial equations with the use of linear algebra methods. His research interests are in the fields of (numerical) linear algebra, polynomial algebra, system identification and machine learning.

List of Publications

Batselier K., Dreesen P., De Moor B., "A geometrical approach to finding multivariate approximate LCMs and GCDs", *Linear Algebra and its Applications*, vol. 438, no. 9, May 2013, pp. 3618-3628.

Batselier K., Dreesen P., De Moor B., "The Geometry of Multivariate Polynomial Division and Elimination", *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, Feb. 2013, pp. 102-125.

Falck T., Dreesen P., De Brabanter K., Pelckmans K., De Moor B., Suykens J.A.K., "Least-Squares Support Vector Machines for the Identification of Wiener-Hammerstein Systems", *Control Engineering Practice* , vol. 20, no. 11, Nov. 2012, pp. 1165-1174.

Geebelen D., Batselier K., Dreesen P., Signoretto M., Suykens J.A.K., De Moor B., Vandewalle J., "Joint Regression and Linear Combination of Time Series for Optimal Prediction", in *Proc. of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, Brugge, Belgium, Apr. 2012.

Dreesen P., Batselier K., De Moor B., "Weighted/Structured Total Least Squares Problems and Polynomial System Solving", in *Proc. of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012)*, Brugge, Belgium, Apr. 2012, pp. 351-356.

Batselier K., Dreesen P., De Moor B., "Prediction Error Method Identification is an Eigenvalue Problem", in *Proc. of the 16th IFAC Symposium on System Identification (SYSID 2012)*, Brussels, Belgium, Jul. 2012, pp. 221-226.

Dreesen P., Batselier K., De Moor B., "Back to the Roots: Polynomial System Solving, Linear Algebra, Systems Theory", in *Proc. of the 16th IFAC Symposium on System Identification (SYSID 2012)*, Brussels, Belgium, Jul. 2012, pp. 1203-1208.

Batselier K., Dreesen P., De Moor B., “Maximum Likelihood Estimation and Polynomial System Solving”, in *Proc. of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2012 (ESANN 2012)*, Brugge, Belgium, Apr. 2012.

De Brabanter K., Dreesen P., Karsmakers P., Pelckmans K., De Brabanter J., Suykens J.A.K., De Moor B., “Fixed-Size LS-SVM Applied to the Wiener-Hammerstein Benchmark”, in *Proc. of the 15th IFAC Symposium on System Identification (SYSID 2009)*, Saint-Malo, France, Jul. 2009, pp. 826-831.

Dreesen P., De Moor B., “Polynomial Optimization Problems are Eigenvalue Problems”, in *Chapter 4 of Model-Based Control – Bridging Rigorous Theory and Advanced Technology*, (Van den Hof P.M.J., Scherer C., and Heuberger P.S.C., eds.), Springer , 2009, pp. 49–68.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING ESAT/STADIUS
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10
B-3001 Heverlee
philippe.dreesen@esat.kuleuven.be
http://www.esat.kuleuven.be