Arjeh M. Cohen   Hans Cuypers
Hans Sterk (Eds.)

Some Tapas of Computer Algebra

With 20 Figures

Springer

Arjeh M. Cohen
Hans Cuypers
Hans Sterk

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
NL-5600 MB Eindhoven
The Netherlands

e-mail
Cohen: amc@win.tue.nl
Cuypers: hansc@win.tue.nl
Sterk: sterk@win.tue.nl

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Some tapas of computer algebra / Arjeh M. Cohen ... (ed.). - Berlin ; Heidelberg ; New York ;
Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 1999
(Algorithms and computation in mathematics ; Vol. 4)
ISBN 3-540-63480-0

Mathematics Subject Classification (1991): 13-01, 13Pxx, 14-01, 14Pxx, 17Bxx, 20Bxx, 20B40, 52B12, 94Bxx, 94B35, 12Hxx, 12A20, 68C20

ISSN 1431-1550

ISBN 3-540-63480-0 Springer-Verlag Berlin Heidelberg New York

Cover design: MetaDesign plus GmbH, Berlin.
Typesetting: Typeset in LaTeX by the editors and reedited by Adam Leinz, Karlruhe, using a Springer LaTeX macro-package.

SPIN 10643224 46/3143 - 5 4 3 2 1 0 - Printed on acid-free paper

Preface

In the years 1994, 1995, two EIDMA minicourses on Computer Algebra were given at the Eindhoven University of Technology by, apart from ourselves, various invited lecturers. (EIDMA is the Research School 'Euler Institute for Discrete Mathematics and its Applications'.) The idea of the courses was to acquaint young mathematicians with algorithms and software for mathematical research and to enable them to incorporate algorithms in their research. A collection of lecture notes was used at these courses.

When discussing these courses in comparison with other kinds of courses one might give in a week's time, Joachim Neubüser referred to our courses as 'tapas'. This denomination underlined that the courses consisted of appetizers for various parts of algorithmic algebra; indeed, we covered such spicy topics as the link between Gröbner bases and integer programming, and the detection of algebraic solutions to differential equations.

As a collection, the notes turned out to have some appeal of their own, which is the main reason why the idea came up of transforming them into book form. We felt however, that the book should be distinguishable from a standard text book on computer algebra in that it retains its appetizing flavour by presenting a variety of topics at an accessible level with a view to recent developments.

Naturally, the book contains a summary of some basic key results, such as Gröbner bases (Chapter 1), the LLL algorithm (Chapter 3), and factorisation (Chapter 4). Despite the variety of topics, the reader will find ample interdependencies. For instance, Chapters 2 and 5 deal with commutative and noncommutative algebra—it is an interesting exercise to see what the commutative counterparts are of the notions defined in Chapter 5. Gröbner basis theory comes back in Chapters 7, 10, and 11 on integer programming, coding theory and decoding, respectively. An outsider may be surprised to see that topics like the sign of a real algebraic number and closed form solutions of differential equations (Chapters 6 and 9) can be dealt with completely algebraically. Chapter 8, on group theoretic algorithms, does not need much of the preceding chapters, but the theory interlocks with coding theory, as becomes apparent in the Projects 6 and 7 on Mathieu groups and Golay codes, respectively.

This brings us to a few words on the seven 'projects'. For those wanting to acquaint themselves somewhat further with part of the material presented in the eleven chapters, these 'projects' have been collected at the end of the book. A project is a coherent body of exercises around a theme, which could serve as a practical session related to one or more chapters. The first project can be dealt with immediately after Chapter 1; Project 2 depends on Chapter 6, Project 3 on Chapters 2 and 6, whereas Projects 4 and 5 add to Chapters 5 and 8, respectively. Finally, the projects already mentioned, *viz.* 6 and 7, depend mainly on Chapters 8 and 10.

We have tried to achieve a uniform set of notes, while preserving some characteristics of the style of the individual authors. For example, in one chapter the exercises occur at the end, whereas, in another, they are interspersed in the text.

We hope that the result may be of use to university teachers composing a course in mathematical aspects of computer algebra as well as to advanced undergraduate students interested in algorithms in algebra.

Eindhoven, April 1998 *A. M. Cohen, H. Cuypers, H. Sterk*

Acknowledgements

The work of Laureano Gonzalez-Vega (Chaps. 2, 6 and Projects 2, 3) and of Maria-Jose Gonzalez-Lopez (Projects 2 and 3) was partially supported by FRISCO (European Union, LTR 21024) and DGICYT PB 95/0563-A (Sistemas de Ecuaciones Algebraicas: Resolución y Aplicaciones). The work of Fabrice Rouillier and Marie-Françoise Roy (Chaps. 2 and 6) was partially supported by FRISCO; the work of Gabor Ivanyos and Lajos Rónyai (Chap. 5) was supported in part by OTKA Grant 016503, AKP Grant 96-350/24, MKM Grant FKFP 0612/1997, and EC Grant ALTEC-KIT.

Contents

List of Contributors

Frits Beukers
University of Utrecht
Department of Mathematics
Budapestlaan 6
Utrecht
The Netherlands
`beukers@math.ruu.nl`

Mario de Boer
Ministerie van Defensie
Kattenburgstraat 7
1018 JA Amsterdam
The Netherlands
`mariodb@worldonline.nl`

Arjeh M. Cohen
Eindhoven University of Technology
Department of Mathematics
and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands
`amc@win.tue.nl`

Hans Cuypers
Eindhoven University of Technology
Department of Mathematics
and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands
`hansc@win.tue.nl`

Gábor Ivanyos
Computer and Automation Institute
Hungarian Academy of Sciences
Lágymányosi u. 11
Budapest H-1111, Hungary
`Gabor.Ivanyos@sztaki.hu`

Maria-Jose Gonzalez-Lopez
Universidad de Cantabria
Departamento de Matemáticas
Estadística y Computación
Facultad de Ciencias
39071 Santander, Spain
`glopez@matesco.unican.es`

Laureano Gonzalez-Vega
Universidad de Cantabria
Departamento de Matemáticas
Estadística y Computación
Facultad de Ciencias
39071 Santander, Spain
`gvega@matesco.unican.es`

Ruud Pellikaan
Eindhoven University of Technology
Deptartment of Mathematics
and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands
`ruudp@win.tue.nl`

Marius van der Put
University of Groningen
Department of Mathematics
Blauwborgje 3
Postbus 800
9700 AV Groningen
The Netherlands
mvdput@math.rug.nl

Tomas Recio
Universidad de Cantabria
Departamento de Matemáticas
Estadística y Computación
Facultad de Ciencias
39071 Santander, Spain
recio@matesco.unican.es

Remko Riebeek
Ministerie van Defensie
Kattenburgstraat 7
1018 JA Amsterdam
The Netherlands
remkor@worldaccess.nl

Lajos Rónyai
Computer and Automation Institute
Hungarian Academy of Sciences
Lágymányosi u. 11
Budapest H-1111, Hungary
lajos@ilab.sztaki.hu

Fabrice Rouillier
INRIA Lorraine
Technopole de Nancy-Brabois
615 rue du Jardin Botanique
F-54600 Villers-les-Nancy, France
Fabrice.Rouillier@loria.fr

Marie-Françoise Roy
Université de Rennes
IRMAR (URA CNRS 305)
Campus de Beaulieu
35042 Rennes cedex, France
costeroy@univ-rennes1.fr

Leonard H. Soicher
Queen Mary and Westfield College
School of Mathematical Sciences
Mile End Road
London E1 4NS, U.K.
L.H.Soicher@maths.qmw.ac.uk

Hans Sterk
Eindhoven University of Technology
Department of Mathematics
and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands
sterk@win.tue.nl

Guadelupe Trujillo
Universidad de Cantabria
Departamento de Matemáticas
Facultad de Ciencias
39071 Santander, Spain
trujillo@matsun1.unican.es

M. Pilar Vélez
Universidad Antonio de Nebrija
Departamento
de Ingeniería Informática
Pirineos, 55
28040 Madrid, Spain
pvelez@dii.unnet.es

Günter M. Ziegler
Technical University Berlin
MA 6-1
Department of Mathematics
10623 Berlin, Germany
ziegler@math.tu-berlin.de

Chapter 1. Gröbner Bases, an Introduction

Arjeh M. Cohen

1. Introduction

Gröbner bases form a core topic of computer algebra and are needed for various subsequent chapters of this book. There are several ways of looking at the famous Buchberger algorithm for constructing Gröbner bases. In this section, we give three interpretations. In the following sections, the Buchberger algorithm and its role according to each interpretation will be discussed in detail.

Rings in this chapter are generally understood to have unit element and to be commutative.

Definition 1.1. A ring R is called *effective* if

- its elements can be described on computer, and equality between two elements can be tested by means of an algorithm,
- its ring operations can be performed by means of algorithms, and
- the solutions of a linear equation $\sum_i a_i x_i = b$ with $a_i, b \in R$ and unknown $x_i \in R$ (in terms of a particular solution and a finite set of generators for the module of all solutions of the corresponding homogeneous equation) can be found algorithmically.

Examples are the integers, the rationals, and algebraic number fields. It may look surprising at first sight that there is an explicit requirement for equality tests between elements. This is due to the fact that often the elements do not have unique representations. The simple case of integers already shows that there is work to be done: if we define integers as strings of digits, we need to equate the elements 01 and 1. In the case of the rationals, the elements are usually represented by pairs of integers (a, b) with $b \neq 0$ (standing for a/b, of course). The equality test between (a, b) and (a', b') is then reduced to the equality test between the integers ab' and $a'b$.

Finite rings are also effective: if R is a ring on n elements ($n < \infty$), it can be presented by a multiplication table and an addition table, that is, (symmetric) matrices whose columns and rows are indexed by the elements of R and whose entries are also filled with elements of R. If the (x, y) entry is z, the interpretation is that the product of x and y is z. Negation, the zero, and the unit element of R can be read off from these tables, and so all ring operations are effective. Finiteness of R also makes solving the linear equation effective: given $a_1, \ldots, a_m \in R^m$ and $b \in R$, a mere exhaustive search for all $(x_1, \ldots, x_m) \in R^m$ satisfying $\sum_{i=1}^m a_i x_i = b$ would solve the equation.

Effectiveness of the field $\mathbb{Q}$ of rational numbers as a ring is also straightforward. Solving the linear equation comes down to expressing one variable as dependent on all others. Solving a linear equation over $\mathbb{Z}$ involves the use of extended gcd's and is less common practice, so we relegate the proof of this fact to an exercise:

Exercise 1.2. Show that $\mathbb{Z}$ is effective.

When we say that a field is *effective*, we do not only mean that it is effective as a ring but also that the inverse of a nonzero element can be found algorithmically. For $\mathbb{Q}$ and for finite fields, this is also clearly satisfied.

Let R be a ring. We are concerned with the polynomial ring $R[\mathcal{X}] = R[X_1, \ldots, X_n]$ in the variables from $\mathcal{X} = \{X_1, \ldots, X_n\}$. If $\mathcal{X}$ consists of the single variable X, we write $R[\mathcal{X}] = R[X]$ and call polynomials in $R[X]$ *univariate*. The ring $R[\mathcal{X}]$ can be identified with the ring $R[X_1, \ldots, X_{n-1}][X_n]$; this means that a polynomial in n variables can be considered as a univariate polynomial whose coefficients are polynomials in $n-1$ variables.

An *ideal* of $R[\mathcal{X}]$ is a subset I of $R[\mathcal{X}]$ such that

- sums of elements of I also belong to I,
- $0 \in I$, and
- the product of an element of I by an element of R also belongs to I.

Examples are $I = \{0\}$, $I = R[\mathcal{X}]$, and $I = \{f \in R[\mathcal{X}] \mid f(0, \ldots, 0) = 0\}$. The last example can be generalized to an arbitrary subset Z of R^n instead of the singleton $\{(0, \ldots, 0)\}$:

$$I(Z) = \{f \in R[\mathcal{X}] \mid \forall_{z \in Z}\ f(z) = 0\}$$

is an ideal of $R[\mathcal{X}]$. This connection between ideals and subsets of R^n is the starting point of algebraic geometry. For more details, see any introduction to algebraic geometry, e.g., [19].

The intersection of a family of ideals of $R[\mathcal{X}]$ is again an ideal of $R[\mathcal{X}]$. Thus, for a subset B of $R[\mathcal{X}]$, we can define (B), the ideal of $R[\mathcal{X}]$ *generated* by $R[\mathcal{X}]$, as the intersection of all ideals containing B. The following theorem shows that ideals of $R[\mathcal{X}]$ can often be described by finite generating sets (sometimes called *bases*). Recall that a ring R is called *Noetherian* if every ideal of R is generated by a finite subset of R. Obviously, fields are Noetherian.

Theorem 1.3 (Hilbert's Basis Theorem). *The ring $R[\mathcal{X}]$ is Noetherian if R is.*

Proof. A proof appears in most introductions to abstract algebra (see, for instance [2] or [3]).

If a ring is Noetherian, and its elements can be properly represented on computer, then so can its ideals, namely by finite lists of ideal generators. Thanks to the theorem, the class of Noetherian rings is a considerably large one. In this context, one might ask whether effectiveness is also inherited from a ring to a polynomial ring over that ring.

Problem 1: For an effective ring R, provide algorithms turning $R[\mathcal{X}]$ into an effective ring.

Suppose that R is effective. Then the elements of $R[\mathcal{X}]$ are clearly representable on computer. Denote by $\mathcal{M}$ the set of all monomials of $R[\mathcal{X}]$. It is a monoid. Besides, it is a linear spanning set for $R[\mathcal{X}]$. (Observe that, if R is a field, $R[\mathcal{X}]$ is a vector space over R with basis $\mathcal{M}$.) If $m \in \mathcal{M}$ then there is a vector $\mathbf{a} \in \mathbb{N}^n$ such that $m = X_1^{a_1} \cdots X_n^{a_n}$, which we often abbreviate to $X^{\mathbf{a}}$. The map $\mathbf{a} \mapsto X^{\mathbf{a}} : \mathbb{N}^n \to \mathcal{M}$ is an isomorphism of monoids. Now each $f \in R[\mathcal{X}]$ is a sum of finitely many *terms*, i.e., elements of the form $cX^{\mathbf{a}}$ with $c \in R$ and $a \in \mathbb{N}^n$. Thus f can be represented on a computer by the list of all such pairs $(c, \mathbf{a})$. Conversely, any set of such pairs with the property that no two have the same monomial $X^{\mathbf{a}}$, uniquely represents a polynomial (the empty set corresponds to the zero polynomial).

Also, the ring operations of $R[\mathcal{X}]$ are easily seen to be effective as well. Thus, the only interesting part of Problem 1 is to describe all solutions of the equation $\sum_i a_i x_i = b$.

Example 1.4. To see that, at first sight, this problem is not trivial, take $a_1 = X_1^2X_2 - 1$, $a_2 = X_1X_2^2 - 1$, and $b = X_1 - X_2$ in $R[\mathcal{X}] = \mathbb{Q}[X_1, X_2]$. A solution to the equation $y_1a_1 + y_2a_2 = b$ is $(y_1, y_2) = (X_2, -X_1)$. To produce a solution, the combination of the polynomials a_1, a_2 in the left-hand side has to decrease the degree of a_1, a_2. There does not seem to be any control over how this may be done. In §6 we will see how the Buchberger algorithm deals with this problem.

For the other two interpretations of the Buchberger algorithm, we need that $R = K$ is an effective field. Then $K[\mathcal{X}]$ is an infinite dimensional vector space over K, and every ideal of $K[\mathcal{X}]$ is a K-linear subspace.

Problem 2: Given a finite set of polynomial equations over K in the variables $\mathcal{X}$, produce a 'triangular form' of the equations, so that one can look for solutions by elimination of variables.

Example 1.5. Compared with the system of polynomial equations for $x_1, x_2 \in K$:

$$a_1(x_1, x_2) = 0, \quad a_2(x_1, x_2) = 0,$$

where a_1, a_2 are as in Example 1.4, the following 'derived system'

$$\begin{array}{rl} x_1a_2(x_1,x_2) - x_2a_1(x_1,x_2): & x_1 - x_2 = 0 \\ (1 + x_1^2x_2)a_2(x_1,x_2) + x_1^3a_2(x_1,x_2): & 1 - x_1^3 = 0 \end{array}$$

is much easier to solve. It has a triangular form in the sense that x_2 is missing from the last equation. Thus, it can be used as a starting point; it says that x_1 is a cube root of unity, $x_1 \in \{1, e^{2\pi i/3}, e^{4\pi i/3}\}$; the first equation is used to express x_2 in the known x_1 (in fact, it tells us that x_1 and x_2 coincide).

To an ideal I of $R[\mathcal{X}]$ we associate an equivalence relation defined by $a \sim b$ if and only if $a - b \in I$. The set of corresponding equivalence classes $R[\mathcal{X}]/I$ is a ring, the *quotient ring* of $R[\mathcal{X}]$ by I. There is a canonical ring morphism from $R[\mathcal{X}]$ to $R[\mathcal{X}]/I$, such that the image of any element of I is 0.

Observe that if $R = K$ is a field, the quotient ring $K[\mathcal{X}]/I$ is also a vector space over K. This gives rise to the idea of representing elements of $K[\mathcal{X}]/I$ by polynomials in a fixed linear subspace S of $K[\mathcal{X}]$ complementary to I. In order to perform the arithmetic of $K[\mathcal{X}]/I$ on elements of S, we need to find, for an arbitrary element $f \in K[\mathcal{X}]$, the unique element $f' \in S$ with $f - f' \in I$.

Problem 3: Given a finite subset B of $K[\mathcal{X}]$, produce an effectively computable K-linear projection map $K[\mathcal{X}] \to S$, where S is a linear subspace of $K[\mathcal{X}]$ isomorphic (as a vector space) to $K[\mathcal{X}]/(B)$, with kernel the ideal (B) of $K[\mathcal{X}]$ generated by B.

Example 1.6. For the subset $B = \{a_1, a_2\}$, with a_1, a_2 as in Example 1.4, the triangular form of the 'equivalent' pair $\{X_1 - X_2,\ 1 - X_1^3\}$ described in Problem 2 suggests the choice of $1, X_1, X_1^2$ as the basis for a complement of (B) in the vector space $R[\mathcal{X}]$. The ring homomorphism that we will find in dealing with this problem will send an arbitrary polynomial $f \in R[\mathcal{X}]$ to

$$(g_0 + g_3 + g_6 + \cdots) + (g_1 + g_4 + g_7 + \cdots)X_1 + (g_2 + g_5 + g_8 + \cdots)X_1^2,$$

where $f(X_1, X_1) = \sum_i g_i X_1^i$. (That is, first replace X_2 by X_1, using that $X_2 - X_1$ lies in the kernel, and then replace all occurrences of X_1^3 by 1, using that $X_1^3 - 1$ lies in the kernel.)

In this chapter, we shall go into each of the above three views of the Buchberger algorithm. To this end we need the notion of a reduction order on monomials; it is introduced in §2. Next, in §3, we present the Buchberger algorithm. We finish with three sections, each dealing with one of the three interpretations, but in the reverse order of their appearance here.

We shall mainly be looking at polynomial rings over a field. For Problem 1, this is a proper restriction of generality, as the theory also works for $R = \mathbb{Z}$. But the presentation is greatly simplified by this assumption.

2. Monomials

Recall that $\mathcal{M}$ denotes the monoid of all monomials of $K[\mathcal{X}]$. We shall use a total order $<$ on $\mathcal{M}$ which is compatible with the monoid structure in the sense that, for all $m, m', m'' \in \mathcal{M}$,

1. $m \geq 1$;
2. if $m' < m''$ then $mm' < mm''$.

A total order with these properties is called a *reduction order*. Note that it refines the division relation: if m divides m' then $m \leq m'$.

Example 2.1. The *lexicographic order* (coming from the identification of $\mathcal{M}$ with $\mathbb{N}^n$). Here $X^{\mathbf{a}} < X^{\mathbf{b}}$ if and only if, for some $k \in \{1, \ldots, n\}$, we have $a_i = b_i$ for all $i < k$ and $a_k < b_k$.
Another example is the *lexicographic total degree order*, that is, order first by total degree, then by lexicographic order, usually taking $X_1 > X_2 > \cdots > X_n$. Here the *total degree* of $X^{\mathbf{a}}$ is $\deg X^{\mathbf{a}} = \sum_{i=1}^{n} a_i$. More generally, an order $<$ is called a *total degree order* if it is a refinement of the partial order $\prec$ given by $X^a \succeq X^b$ if and only if $\deg X^{\mathbf{a}} \geq \deg X^{\mathbf{b}}$.

Exercise 2.2. Prove that, if $n = 1$, the natural order on $\mathbb{N}$ is the only reduction order on $\mathcal{M} = \mathbb{N}$.

Exercise 2.3. Prove that the following order is a reduction order on the monomials of $K[\mathcal{X}]$: $X^a < X^b$ if and only if either $\sum_i a_i < \sum_i b_i$ or $\sum_i a_i = \sum_i b_i$ and there is $k \in \{1, \ldots, n\}$ with $a_k < b_k$ and $a_j = b_j$ for all $j > k$.

Exercise 2.4. Verify that a total degree ordering has the property that, for each $m \in \mathcal{M}$, the set of all monomials less than m is finite. Show that this is not true for a lexicographic order.

Termination of the Buchberger algorithm is based on the fact that the orders involved are well founded. Recall that an order is called *well founded* if each strictly descending chain is finite. Denote by $\mathcal{F}(\mathcal{M})$, or simply $\mathcal{F}$, the collection of all finite subsets of $\mathcal{M}$. For $A \in \mathcal{F}$ and $u \in \mathcal{M}$ write

$$A_{>u} = \{x \in A \mid x > u\}.$$

If B is another member of $\mathcal{F}$, then, since A and B are finite, the finite (nonempty) subset $(A \cup B) \setminus (A \cap B)$ has a (unique) maximal element u with respect to $<$. We write $A >_{\mathcal{F}} B$ if u belongs to A and $A <_{\mathcal{F}} B$ otherwise. Thus, $A <_{\mathcal{F}} B$ means that there exists $u \in B \setminus A$ such that $A_{>u} = B_{>u}$.

Proposition 2.5. *The following holds for $\mathcal{M}$.*

1. *Each reduction order on $\mathcal{M}$ is well founded.*
2. *If $<$ is a total order on $\mathcal{M}$, then the relation $<_{\mathcal{F}}$ is a total order on $\mathcal{F}$.*
3. *If, moreover, $<$ is well founded, then so is $<_{\mathcal{F}}$.*

Proof. 1. Suppose that there exists an infinite chain $m_1 > m_2 > \cdots$ in $\mathcal{M}$. Then $n > 1$, for otherwise $n = 1$ and $<$ is the usual order on $\mathbb{N}$, which is obviously well founded. One way to obtain a contradiction is to derive an infinite strictly descending chain in $\mathbb{N}^{n-1}$ from the given one in $\mathcal{M} \cong \mathbb{N}^n$.

Another approach is to use Theorem 1.3. Here, we consider the ideals

$$I_j = \{m_1, \ldots, m_j\}K[\mathcal{X}] \quad \text{for} \quad j = 1, 2, \ldots,$$

where, for a subset A of $K[\mathcal{X}]$, we write $A \cdot K[\mathcal{X}]$ or (A) for the ideal generated by A; thus $A \cdot K[\mathcal{X}]$ is an alternative notation for (A), which stresses the dependence on the ring $K[\mathcal{X}]$. These ideals form an ascending chain. Since $K[\mathcal{X}]$ is Noetherian, the chain terminates, so there is $N \in \mathbb{N}$ with $m_j \in I_N$ for all $j \geq N$. Take $j > N$. Then, as $m_j \in I_N$, there is $i \leq N$ with $m_i | m_j$. Hence $m_i \leq m_j$, contradicting the assumption that $m_i > m_j$ for $i < j$.

2. Straightforward.

3. For $F \in \mathcal{F}$, we denote by max F the maximal element in F with respect to $<$. Assume that $F_1 >_{\mathcal{F}} F_2 >_{\mathcal{F}} \cdots$ is an infinite strictly descending sequence in $\mathcal{F}(\mathcal{M})$. Then $\max F_1 \geq \max F_2 \geq \cdots$ is a weakly descending infinite chain in $\mathcal{M}$. By 1., there are a number N_1 and a monomial f_1 such that $\max F_k = f_1$ for all $k > N_1$. For each $j \geq 1$, set $F_j^{(1)} = F_{N_1+j} \setminus \{f_1\}$. Now $F_1^{(1)} >_{\mathcal{F}} F_2^{(1)} >_{\mathcal{F}} \cdots$ is again infinite and descending. Thus, there are a number N_2 and a monomial f_2 such that $\max F_k^{(1)} = f_2$ for all $k > N_2$, and so on. In this way, a descending sequence of monomials $f_1 > f_2 > \cdots$ arises. As such a sequence cannot be infinite, we find a number M such that $f_i = f_M$ for all $i > M$. But that implies that there exists a number ℓ such that $F_i = F_\ell$ for all $i > \ell$, a contradiction.

Exercise 2.6. Extend Part 3 of Proposition 2.5 in two different ways for a reduction order $<$ on $\mathcal{M}$:

1. Let B be a finite totally ordered set and $\mathcal{G}$ the set of all maps $B \to \mathcal{M}$. Prove that the relation $<_{\mathcal{G}}$ on $\mathcal{G}$ given by
$$g <_{\mathcal{G}} g' \iff \exists_{b \in B}\ g_b < g'_b \text{ and } \forall_{c > b}\ g_c = g'_c,$$
for $g, g' \in \mathcal{G}$, is a well-founded order on $\mathcal{G}$.
2. Let $\mathcal{H}$ be the collection of all finite multisets in $\mathcal{M}$ (that is, all maps $\mathcal{M} \to \mathbb{N}$ with nonzero image in only finitely many members of $\mathcal{M}$). Prove that the relation $<_{\mathcal{H}}$ on $\mathcal{H}$ given by
$$g <_{\mathcal{H}} g' \iff \exists_{m \in \mathcal{M}}\ g(m) < g'(m) \text{ and } \forall_{n > m}\ g(n) = g'(n),$$
for $g, g' \in \mathcal{H}$, is a well-founded order on $\mathcal{H}$.

We shall use these orders to compare (sets of) polynomials.

Definition 2.7. To single out the highest monomial and coefficient from a polynomial $f \in K[\mathcal{X}]$, we set

1. $\mathrm{lm}(f) = \max\{m \in \mathcal{M} \mid f_m \neq 0\}$, with the convention that $\mathrm{lm}(0) = -\infty$;
2. $\mathrm{lc}(f)$ is the coefficient of the monomial $\mathrm{lm}(f)$ of f, with the convention that $\mathrm{lc}(0) = 0$;
3. $\mathrm{lt}(f) = \mathrm{lc}(f)\,\mathrm{lm}(f)$.

The symbols lm, lc, lt stand for *leading monomial*, *leading coefficient* and *leading term*, respectively. For $f \in K[\mathcal{X}]$, we shall write $\mathcal{M}_f$ to denote the set of monomials occurring in f.

Exercise 2.8. Prove that, for $f, g \in K[\mathcal{X}]$, we have $\mathrm{lt}(fg) = \mathrm{lt}(f)\mathrm{lt}(g)$ and $\mathrm{lm}(f+g) \leq \max(\mathrm{lm}(f), \mathrm{lm}(g))$. Give an example to show that equality does not always hold in the latter inequality.

Exercise 2.9. Instead of $K[\mathcal{X}]$, one can study more general rings and still construct Gröbner bases using reduction orders. A good setting is the 'monoid ring' $R \cdot M = \bigoplus_{m \in M} Rm$, where M is a monoid admitting a reduction order and R is an arbitrary ring. But not all monoids admit reduction orders. Show that a nontrivial finite cyclic monoid does not admit a reduction order. How about the additive monoid $\mathbb{Z}$?

Exercise 2.10. Many examples of reduction orderings on $\mathcal{M}$ are obtained as follows: Starting from any reduction order $<$ and a vector $\mathbf{c} \in \mathbb{N}^n$, a new reduction order $<_c$ is obtained by requiring that $X^{\mathbf{a}} <_c X^{\mathbf{b}}$ if and only if either $\mathbf{a} \cdot \mathbf{c} < \mathbf{b} \cdot \mathbf{c}$ or $\mathbf{a} \cdot \mathbf{c} = \mathbf{b} \cdot \mathbf{c}$ and $X^{\mathbf{a}} < X^{\mathbf{b}}$. Here $\mathbf{a} \cdot \mathbf{c} = \sum_{i=1}^{n} a_i c_i$. If $\mathbf{c}$ is the all-one vector, the order is a total degree order.

3. The Buchberger Algorithm

In the sequel, we shall often assume that $R = K$ is an effective field. One can work with less drastic restrictions, but life is simpler this way.

Let B be a finite set of polynomials in $K[\mathcal{X}]$ and $f \in K[\mathcal{X}]$. The residue class $f + (B)$ in $K[\mathcal{X}]/(B)$ consists of all polynomials of the form $f + \sum_{b \in B} g_b b$, where $g_b \in K[\mathcal{X}]$. We are after a 'small' canonical representative of $f + (B)$, that is, an element of the residue class which is as small as possible in some sense and hopefully unique. In order to specify what small is, we need to measure polynomials. Different measures lead to different algorithms. One possible measure is $\mathrm{lm}(g)$ for a polynomial g. The following routine makes $\mathrm{lm}(g)$, for g running through $f + (B)$, as small as possible 'at first sight'. This is achieved by subtracting suitable multiples of polynomials from B.

```
Reduce(B, f) =
            $J := \{b \in B \mid \mathrm{lm}(b) \,|\, \mathrm{lm}(f)\}$;
            if $J \neq \emptyset$
            then choose $b \in J$;
              return Reduce($B, f - (\mathrm{lt}(f)/\mathrm{lt}(b))b$)
            else return $f$
            fi.
```

Note that the algorithm terminates after a finite number of steps, since the second argument in the recursive call of *Reduce* has a leading monomial which is smaller than $\mathrm{lm}(f)$ (so that well-foundedness of $<$ does the job). For given $B \subset K[\mathcal{X}]$ and $f \in K[\mathcal{X}]$, the output of *Reduce*(B, f) is a polynomial $g \in f+(B)$ with $\mathrm{lm}(b) \not| \mathrm{lm}(g)$ for every $b \in B$. This polynomial g has the property that its leading monomial cannot be 'reached' by a leading monomial from B.

Example 3.1. The routine *Reduce* is not fully determined by what we have written, as we shall see by taking $B = \{X^2Y - 1, XY^2 - 1\}$. (Here, as more often, we replace X_1 and X_2 by X and Y, respectively, to improve readability.) Then, for $f = X^2Y^2$, we obtain $Reduce(B, f) = f - X(XY^2 - 1) = X$ or $Reduce(B, f) = f - Y(X^2Y - 1) = Y$ according as we choose $b = X^2Y - 1$ or $b = XY^2 - 1$ in B with $\mathrm{lm}(b) \mid \mathrm{lm}(f)$.

Definition 3.2. A polynomial $f \in K[\mathcal{X}]$ is said to be *reduced* (modulo B and $<$) if $f = Reduce(B, f)$. Moreover, $g \in K[\mathcal{X}]$ is said to be a *reduced form of $f \in K[\mathcal{X}]$* (with respect to B and $<$) if $g = Reduce(B, f)$ (for at least some fully specified version of the algorithm *Reduce*).

Remark 3.3. In the literature, the reduced form of f is also referred to as the *remainder* of f with respect to B. This terminology nicely points out that reduction is a generalisation of the familiar univariate polynomial division with remainder.

Remark 3.4. The output $Reduce(B, f) = 0$ implies that

$$f = \sum_{b \in B} g_b b \in (B) \text{ with } \mathrm{lm}(g_b b) \leq \mathrm{lm}(f) \text{ for all } b \in B. \tag{3.1}$$

An obstruction to membership testing ('does f belong to (B)?') using *Reduce* is that it may occur that, although $f \in (B)$ is reduced modulo B, there is no expression of f satisfying (3.1). For instance, with B as in Example 3.1, the polynomial $f = X - Y$ is in (B), as $f = Y(X^2Y - 1) - X(XY^2 - 1) = Yb_1 - Xb_2$; but $\mathrm{lm}(Yb_1) = \mathrm{lm}(Xb_2) = X^2Y^2 > \mathrm{lm}(f)$, and there is no expression $f = \sum_{b \in B} g_b b$ with $\mathrm{lm}(g_b b) \leq \mathrm{lm}(f)$ for all $b \in B$.

Definition 3.5. As mentioned before, other measures on polynomials may lead to different smallest representative polynomials of $f + (B)$. If we take $\mathcal{M}_g$, the (finite) set of all monomials of g, to measure $g \in K[\mathcal{X}]$, then an important variation of *Reduce* arises. Now we subtract from g a multiple of $b \in B$ by a term t whenever $\mathrm{lt}(tb)$ is a term of g (not necessarily $\mathrm{lt}(g)$):

```
StronglyReduce(B, f) =
        J := {(b, m) ∈ B × M_f | lm(b) | m};
        if J ≠ ∅
        then choose (b, m) ∈ J;
        return $StronglyReduce(B, f - (f_m m/\mathrm{lt}(b))b)$
    else return $f$
    fi.
```

Observe that $\mathcal{M}_f$ decreases at each recursive call with respect to $<_{\mathcal{F}}$ as in Proposition 2.5. This justifies the remark that *StronglyReduce* attempts to minimize $\mathcal{M}_g$ for $g \in f + (B)$. We shall call $f \in K[\mathcal{X}]$ *strongly reduced* with respect to B and the fixed reduction order $<$, if the algorithm *StronglyReduce*, when applied to (B, f), yields f itself. In other words, if there is no monomial $m \in \mathcal{M}_f$ that is a multiple of the leading monomial of a member of B.

Exercise 3.6. Consider the algorithm *StronglyReduce* defined above.

1. Show that *StronglyReduce* terminates. Hint: use Part 3 of Proposition 2.5 applied to the sequence $\mathcal{M}_f$ for f the subsequent second arguments in the recursive calls of the routine.
2. Show that, for given $B \subset K[\mathcal{X}]$ and $f \in K[\mathcal{X}]$, $StronglyReduce(B, f)$ is a polynomial $g \in f + (B)$ with $\mathrm{lm}(b) \nmid m$ for $b \in B$ and $m \in \mathcal{M}_g$.
3. Take $f = XY$ and $g = X - Y \in R[X, Y]$ and let $<$ be the lexicographic order on $\mathcal{M}$ with $X > Y$. Verify that $StronglyReduce(\{f, g\}, f)$ may produce both 0 and Y^2.

Definition 3.7. Fix a reduction order on $\mathcal{M}$. In order to formulate the Buchberger algorithm, we need the notion of S-polynomial of $f, g \in K[\mathcal{X}]$. It is the most efficient $K[\mathcal{X}]$-linear combination of f and g that has a leading term of which we do not a priori know that it can be reduced modulo $\{f, g\}$ (see Exercise 3.8 for an exception to this rule). This is done by 'matching' the leading terms for f and g by multiplication with suitable terms and next subtracting these multiples of f and g from each other. For instance, if $<$ is a total degree reduction order, $f = 6X^2Y + 3X + 2$, and $g = 3XY^2 - Y - X$, then their S-polynomial is $(1/6)Yf - (1/3)Xg = (2X^2 + 5XY + 2Y)/6$. The formal definition of *S-polynomial* is

$$S(f, g) = \frac{\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))}{\mathrm{lt}(f)} f - \frac{\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))}{\mathrm{lt}(g)} g$$

if $fg \neq 0$ and 0 otherwise. (Here, lcm stands for *least common multiple.*) Cancellation of the leading terms implies

$$\mathrm{lm}(S(f, g)) < \mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g)). \tag{3.2}$$

Observe that the S-polynomial of f and g is a member of $\{f, g\}K[\mathcal{X}]$. For instance, in the above example, $S(X^2Y - 1, XY^2 - 1) = X - Y$. The fact that $X - Y$ belongs to the ideal (B) also indicates that the difference of the two outcomes of $Reduce(B, f)$ as computed in Example 3.1 lies in (B).

Exercise 3.8. Suppose that $f, g \in K[\mathcal{X}]$ satisfy $\gcd(\mathrm{lm}(f), \mathrm{lm}(g)) = 1$. Prove that $Reduce(\{f, g\}, S(f, g)) = 0$.

In the Buchberger algorithm, the S-polynomial is used to find new members of (B) from two polynomials in B which do not yet reduce to 0 mod B. If the reduction of the result is nonzero, then we add it to B. Here is the Buchberger algorithm in its simplest form.

```
$GroebnerBasis(B)$ =
        $\mathcal{P} :=$ {unordered pairs from $B$} ;
        while $\mathcal{P} \neq \emptyset$ do
          choose $\{f, g\} \in \mathcal{P}$;
          $\mathcal{P} := \mathcal{P} \setminus \{f, g\}$;
          $\mathrm{c} := Reduce(B, S(f, g))$;
          if $\mathrm{c} \neq 0$
          then $B := B \cup \{\mathrm{c}\}$;
            $\mathcal{P} := \mathcal{P} \cup \{\{b, \mathrm{c}\} \mid b \in B\}$ ;
          fi;
        od; return $B$.
```

Termination is a consequence of the fact that the sequence $\{\mathrm{lm}(b) \mid b \in B\}K[\mathcal{X}]$ of ideals of $K[\mathcal{X}]$ for subsequent values of B, produced in the course of the algorithm, strictly ascends. Since $K[\mathcal{X}]$ is Noetherian, this chain must be finite, and so the addition to B stops after a finite number of times. If there are no additions to B, the finite set $\mathcal{P}$ will be exhausted, whence termination.

It is very hard to provide a good explicit upper bound on these chains of increasing B's; bad examples are known.

For simplicity of presentation, we have not taken into account that, at some stages during the algorithm, one can actually discard some members of B, so that B does not necessarily grow as wildly as suggested. Also, Exercise 3.8 can be used to eliminate certain pairs from $\mathcal{P}$ before starting to compute the corresponding S-polynomial.

The result of a *GroebnerBasis* computation with input B is a particularly nice generating set for (B) in that it is a Gröbner basis in the following sense. (The proof of this observation will be given in Corollary 3.12.)

Definition 3.9. A finite subset B of $K[\mathcal{X}]$ is called a *Gröbner basis* if, for each polynomial $f \in (B)$, the zero polynomial is a reduced form of f modulo B. If so, and if $I = (B)$, we also say that B is a *Gröbner basis of I.*

Theorem 3.10 (Gröbner Basis Characterisation). *The following statements concerning a finite subset B of $K[\mathcal{X}]$ with respect to a reduction ordering $<$ on $\mathcal{M}$ are equivalent.*

1. B is a Gröbner basis;
2. $S(b, c)$ reduces to 0 modulo B for each pair $b, c \in B$;
3. $\{\mathrm{lm}(f) \mid f \in (B)\}K[\mathcal{X}] = \{\mathrm{lm}(b) \mid b \in B\}K[\mathcal{X}]$;
4. $\{\mathrm{lt}(f) \mid f \in (B)\}K[\mathcal{X}] = \{\mathrm{lt}(b) \mid b \in B\}K[\mathcal{X}]$.

Clearly, 3 and 4 are equivalent. We have added 4 because (other than 3) it is the criterion that can also be used as a definition of Gröbner basis in case K is not a field. Note that, due to Criterion 2, it can be effectively verified whether B is a Gröbner basis; in fact, B is a Gröbner basis if and only if the algorithm *GroebnerBasis* above leaves B unaltered. Besides, it does not matter which of the two reductions is applied in Criterion 2: it is equivalent to both $Reduce(B, f) = 0$ and $StronglyReduce(B, f) = 0$ (and to (3.1) in Remark 3.4).

Proof. Actually, only one implication is hard to prove: 2 ⇒ 3.
1 ⇒ 2 is trivial as $S(b, c) \in (B)$.
2 ⇒ 3. Suppose $f \in (B)$. We will show that $\mathrm{lm}(f) \in (\{\mathrm{lm}(a) \mid a \in B\})$. This will imply the nontrivial inclusion in 3. Since K is a field, we may (and shall) assume, without loss of generality, that all polynomials b in B are monic, that is, have $\mathrm{lc}(b) = 1$. By assumption there is a map $g : B \to K[\mathcal{X}]$, $a \mapsto g_a$ with

$$f = \sum_{a \in B} g_a a.$$

To this expression for f we associate the multiset $A_g : \mathcal{M} \to \mathbb{N}$ given by

$$A_g = m \mapsto \#\{a \in B \mid \mathrm{lm}(g_a a) = m\}.$$

We assume that g is chosen such that A_g is minimal with respect to the order $<_{\mathcal{H}}$ on $\mathcal{H}$, the collection of finite multisets $\mathcal{M} \to \mathbb{N}$, defined in Part 2 of Exercise 2.6.

Obviously, $\max A_g \geq \mathrm{lm}(f)$. We show that we can achieve equality. Suppose, to this end, $\max A_g > \mathrm{lm}(f)$. Then there must be $b, c \in B$ with $\mathrm{lm}(g_b b) = \mathrm{lm}(g_c c)$ attaining this maximum. (For the monomial does not occur in the left-hand side of the equation $f = \sum_{a \in B} g_a a$ and so must occur at least twice in the right hand side.)

As $Reduce(B, S(b, c)) = 0$, in view of Inequality (3.2), there exist $h_a \in K[\mathcal{X}]$ $(a \in B)$ with

$$S(b, c) \quad = \quad \sum_{a \in B} h_a a \ \text{ and } \forall_{a \in B}\ \mathrm{lm}(h_a a) < \mathrm{lcm}(\mathrm{lm}(b), \mathrm{lm}(c)). \qquad (3.3)$$

We shall use this expression of $S(b, c)$ to construct a new coefficient system $g' : B \to K[\mathcal{X}]$ with $A_{g'} <_{\mathcal{H}} A_g$ and $f = \sum_{a \in B} g'_a a$, thus obtaining a contradiction with the minimality of A_g.

It follows from $\mathrm{lm}(g_b b) = \mathrm{lm}(g_c c)$ that $\mathrm{lcm}(\mathrm{lm}(b), \mathrm{lm}(c))$ divides $\mathrm{lm}(g_b b)$. Let t be the quotient term

$$t = \frac{\mathrm{lt}(g_b b)}{\mathrm{lcm}(\mathrm{lm}(b), \mathrm{lm}(c))}.$$

Then

$$\mathrm{lt}(g_b) = \frac{t\, \mathrm{lt}(c)}{\mathrm{gcd}(\mathrm{lm}(b), \mathrm{lm}(c))},$$

so,

$$\begin{aligned} g_b b &= \mathrm{lt}(g_b)b + (g_b - \mathrm{lt}(g_b))b \\ &= t\frac{\mathrm{lt}(c)b - \mathrm{lt}(b)c}{\mathrm{gcd}(\mathrm{lm}(b), \mathrm{lm}(c))} + t\frac{\mathrm{lt}(b)c}{\mathrm{gcd}(\mathrm{lm}(b), \mathrm{lm}(c))} + (g_b - \mathrm{lt}(g_b))b \\ &= tS(b, c) + \frac{\mathrm{lt}(bg_b)c}{\mathrm{lt}(c)} + (g_b - \mathrm{lt}(g_b))b \\ &= \sum_{a \in B} th_a a + \mathrm{lc}(g_b)\mathrm{lm}(g_c)c + (g_b - \mathrm{lt}(g_b))b, \end{aligned}$$

whence $f = \sum g'_a a$, with

$$\begin{aligned} g'_b &= th_b + g_b - \mathrm{lt}(g_b), \\ g'_c &= g_c + th_c + \mathrm{lc}(g_b)\mathrm{lm}(g_c), \text{ and} \\ g'_a &= g_a + th_a \text{ for } a \in B \setminus \{b, c\}. \end{aligned}$$

In view of (3.1) in Remark 3.4, Inequality (3.2), and the definition of t, we derive that $g' : B \to K[\mathcal{X}]$ satisfies

$$\begin{aligned} \mathrm{lm}(g'_b b) &\le \max(\mathrm{lm}(th_b b), \mathrm{lm}(g_b b - \mathrm{lt}(g_b)b)) < \mathrm{lm}(g_b b) = \max A_g, \\ \mathrm{lm}(g'_c c) &\le \max(\mathrm{lm}(g_c c), \mathrm{lm}(th_c c)) \\ &\le \mathrm{lm}(g_c c) = \max A_g, \text{ and, for } a \in B \setminus \{b, c\}, \\ \mathrm{lm}(g'_a a) &\le \max(\mathrm{lm}(g_a a), \mathrm{lm}(th_a a)) \le \mathrm{lm}(g_b b) = \max A_g, \end{aligned}$$

with equality only if $\mathrm{lm}(g_a a) = \mathrm{lm}(g_b b)$. This implies $A_{g'} <_{\mathcal{H}} A_g$, a contradiction with the minimality of A_g. Thus, we must have $\max A_g = \mathrm{lm}(f)$. But then $\mathrm{lm}(f) = \mathrm{lm}(g_b b)$ for some $b \in B$, so $\mathrm{lm}(f) \in \{\mathrm{lm}(b) \mid b \in B\}K[\mathcal{X}]$, and we are done.

4 ⇒ 1. Assume that Criterion 4 holds and let $f \in (B)$. Without loss of generality, we may assume $f = \mathit{Reduce}(B, f)$. Suppose $f \neq 0$. Then, since $\mathrm{lt}(f) \in \{\mathrm{lt}(b) \mid b \in B\}K[\mathcal{X}]$, there are a nonzero $b \in B$ and $g \in K[\mathcal{X}]$ with $\mathrm{lt}(f) = \mathrm{lt}(gb)$. But then $\mathrm{lm}(f - gb) < \mathrm{lm}(f)$, so f can be reduced, a contradiction. Consequently, $f = 0$, proving Criterion 1.

Corollary 3.11. *Suppose that I is an ideal of $K[\mathcal{X}]$ and B is a finite subset of I. If*

$$\{\mathrm{lt}(b) \mid b \in B\}K[\mathcal{X}] = \{\mathrm{lt}(f) \mid f \in I\}K[\mathcal{X}],$$

then B is a Gröbner basis of I.

Proof. Clearly, $(B) \subseteq I$. Suppose $f \in I$. We want to establish $f \in (B)$. Without loss of generality, we may assume $f = \mathit{Reduce}(B, f)$. Arguing as in '4 ⇒ 1' of the proof of the above theorem, we find that $f = 0$, proving $f \in (B)$.

Corollary 3.12. *The result of the algorithm GroebnerBasis applied to B is a Gröbner basis B' with $(B) = (B')$.*

Proof. Only elements of (B) are added to B in order to get B', so clearly, $(B') = (B)$. At the end of the algorithm, Part 2 of Theorem 3.10 is satisfied and so B' is a Gröbner basis.

Remark 3.13. Corollary 3.11 is extremely useful. For instance, it shows that we may throw away members b from a Gröbner basis B that do not contribute to the so-called initial ideal $\{\mathrm{lt}(a) \mid a \in B\}K[\mathcal{X}]$, in the sense that

$$\{\mathrm{lt}(a) \mid a \in B\}K[\mathcal{X}] = \{\mathrm{lt}(a) \mid a \in B \setminus \{b\}\}K[\mathcal{X}].$$

Thus, one can trim down a Gröbner basis to a *minimal Gröbner basis*, that is, one from which removal of an element would always lead to a strictly smaller ideal. In fact, we can do better and even construct a unique Gröbner basis (for a fixed reduction ordering), see Theorem 3.16.

Example 3.14. We first compute the Gröbner basis of $B = \{b_1, b_2\}$ with $b_1 = X^2Y - 1$ and $b_2 = XY^2 - 1$ as in Example 3.1 with respect to the lexicographic order in which $X > Y$. We have already seen that

$$b_3 = \mathit{Reduce}(B, S(b_1, b_2)) = X - Y.$$

So the next values of B and $\mathcal{P}$ are

$$\begin{aligned} B &= \{X^2Y - 1, XY^2 - 1, X - Y\}, \\ \mathcal{P} &= \{\{X^2Y - 1, X - Y\}, \{XY^2 - 1, X - Y\}\}. \end{aligned}$$

Now picking the first pair from $\mathcal{P}$, we find $c = 0$, so we can continue with the second pair. This produces $c = Y^3 - 1$, so we must append $b_4 = Y^3 - 1$ to B. Moreover, updating $\mathcal{P}$ yields

$$\mathcal{P} = \{\{X^2Y - 1, Y^3 - 1\}, \{X - Y, Y^3 - 1\}, \{XY^2 - 1, Y^3 - 1\}\},$$

which looks pretty horrible. Actually, now the S-polynomials of all pairs in $\mathcal{P}$ reduce to zero, so we are done. Thus the Gröbner basis is $\{b_1, b_2, b_3, b_4\}$. Due to Remark **3.13**, $\{b_3, b_4\}$ is also a Gröbner basis of the same ideal.

Definition 3.15. Let $B \subset K[\mathcal{X}]$ be a finite subset of $K[\mathcal{X}]$. Then B is called a ***reduced Gröbner basis*** if it is a Gröbner basis and if each $b \in B$ has $\mathrm{lc}(b) = 1$ and is strongly reduced with respect to $B \setminus \{b\}$ (in formula, $b = \mathit{StronglyReduce}(B \setminus \{b\}, b)$).

The strength of this notion will be clear from the following result.

Theorem 3.16. *For a given reduction order, each ideal of $K[\mathcal{X}]$ has a unique reduced Gröbner basis.*

Thus, one can tell if two ideals are equal by computing their reduced Gröbner bases.

Proof. First we prove existence. For each $b \in B$, the condition on $\mathrm{lc}(b) = 1$ can obviously be satisfied at any time by dividing b by $\mathrm{lc}(b)$. Thus, we can, and will, assume that the members of B are *monic*, i.e., have leading coefficient equal to 1. By Corollary 3.12, given a finite set of generators of an ideal, we can find a Gröbner basis B for that ideal.

If $b \in B$ is not strongly reduced with respect to $B \setminus \{b\}$, there is $c \in B$, $c \neq b$, with $\mathrm{lm}(c) \mid m$, where $m \in \mathcal{M}_b$. Write $b' = b - b_m \mathrm{lm}(b) c/\mathrm{lt}(c)$. This is the result of the first step in *StronglyReduced*$(B \setminus \{b\}, b)$ and $\mathcal{M}_{b'} < \mathcal{M}_b$. Writing $\mathcal{M}(B)$ for the multiset:

$$\mathcal{M}(B) = (m \in \mathcal{M} \mapsto \#\{b \in B \mid b_m \neq 0\}),$$

this gives $\mathcal{M}(B) >_{\mathcal{H}} \mathcal{M}(B \cup \{b'\} \setminus \{b\})$, with the notation of Exercise 2.6. By transitivity of $<_{\mathcal{H}}$, we find $\mathcal{M}(B) = \mathcal{M}(B')$ if B' is obtained from B by substituting a strong reduction of b by $B \setminus \{b\}$ (and removing 0 from the result). Hence, well foundedness of $<_{\mathcal{H}}$ (cf. 2.6) gives termination of the algorithm replacing each strongly reducible $b \in B$ with respect to $B \setminus \{b\}$ by $c :=$ *StronglyReduce*$(B \setminus \{b\}, b)$ if $c \neq 0$ and removing it from B if $c = 0$. After termination, all elements $b \in B$ are strongly reduced with respect to $B \setminus \{b\}$, and so we have obtained a reduced Gröbner basis.

Suppose B and C are both reduced Gröbner bases of (B). Let $b \in B$. Then there is $c \in C$ such that $\mathrm{lt}(c) = \mathrm{lm}(c) \mid \mathrm{lm}(b) = \mathrm{lt}(b)$. On the other hand, there is $b' \in B$ such that $\mathrm{lt}(b') \mid \mathrm{lt}(c)$. But then $\mathrm{lt}(b') \mid \mathrm{lt}(b)$, so the fact that B is reduced implies $b = b'$. Hence $\mathrm{lt}(c) = \mathrm{lt}(b)$. We have shown that for every member of B there is an element of C with the same leading term. By symmetry, we have the same with the roles of B and C interchanged. This observation already implies that B and C have the same cardinality.

Now take $b \in B \setminus C$ such that $\mathrm{lm}(b)$ is minimal. As argued above, there is $c \in C$ such that $\mathrm{lt}(b) = \mathrm{lt}(c)$. Consider $b - c$. It belongs to (B) and $\mathrm{lm}(b - c) < \mathrm{lm}(b)$, so, by minimality of $\mathrm{lm}(b)$, there are f_d $(d \in B \cap C)$ such that $b - c = \sum_{d \in B \cap C} f_d d$ with $\mathrm{lm}(f_d d) < \mathrm{lm}(b)$. We want to show that the sum is empty. Suppose that $f_d \neq 0$ for some $d \in B \cap C$. Then there is a monomial $t \in \mathcal{M}_b \cup \mathcal{M}_c$ such that $\mathrm{lm}(d) \mid \mathrm{lm}(t)$, contradicting that B and C are reduced Gröbner bases. Hence $\sum_d f_d d = 0$ and $b = c$, which conflicts with $b \notin C$. We have shown that $B \subseteq C$. By symmetry, we conclude that B coincides with C.

Exercise 3.17. Give a simple example to show that the reduced Gröbner basis depends on the reduction order.

4. Standard Monomials

This section is devoted to Problem 3: produce an effectively computable K-linear projection map from $K[\mathcal{X}]$ onto a complement of (B) in $K[\mathcal{X}]$. We shall first motivate why we want such a map.

By Hilbert's Basis Theorem 1.3, a quotient ring of $K[\mathcal{X}]$ is of the form $K[\mathcal{X}]/(B)$ for a finite subset B of $K[\mathcal{X}]$. Thus, the projection of $K[\mathcal{X}]$ onto a complement U of (B) in $K[\mathcal{X}]$ will provide an isomorphism $K[\mathcal{X}]/(B) \cong U$ of K-vector spaces. This means that U can serve as a model for $K[\mathcal{X}]/(B)$. Addition and subtraction of members of U is clear from the vector space structure. The multiplication of two polynomials $f, g \in U$ gives a polynomial fg not necessarily in U, but its projection onto U (with kernel (B)) will provide the unique polynomial in U corresponding to fg. In this manner, the K-algebra operations (testing equality, multiplication, etc.) of $K[\mathcal{X}]/(B)$ are effective whenever the projection onto U is.

Example 4.1. Let $B = \{f\}$ where f is a univariate polynomial in $K[X]$ (with $X = X_1$) of degree m, and let U be the linear subspace of $K[X]$ consisting of all polynomials of degree less than m. Then each element $g + f\,K[X] \in K[X]/(f)$ corresponds to a unique polynomial of U: just take the remainder of division by f, i.e., *Reduce*$(\{f\}, g)$. Using the linear projection map $K[X] \to U$, $g \mapsto$ *Reduce*$(\{f\}, g)$, we can express all arithmetic operations of $K[X]/(f)$, such as addition, multiplication, and even division, in terms of polynomial operations on U.

Exercise 4.2. Suppose $f \in K[X]$ is irreducible of degree m (so that $K[X]/(f)$ is an extension field of K of degree m). Show that the inverse in $K[X]/(f)$ of a nonzero element $g + f\,K[X]$ can be determined by the Extended Euclidean Algorithm. Hint: Find polynomials $u, v \in K[X]$ with $ug + vf = 1$; then u represents the inverse of g.

Definition 4.3. Given a reduction order $<$ on the monomials $\mathcal{M}$ of $K[\mathcal{X}]$, the monomials not divisible by the leading monomial of any member of (B) are called *standard.*

By U we denote the K-linear subspace of $K[\mathcal{X}]$ spanned by all standard monomials. It is a vector space isomorphic to the vector space $K[\mathcal{X}]/(B)$. Recall that $\mathcal{M}_f$ denotes the set of monomials occurring in the polynomial f. One can try and describe the projection $K[\mathcal{X}] \to U$ with kernel (B) as iteratively replacing $f \in K[\mathcal{X}]$ by $g = f - ra$ with $a \in B$ and $r \in K[\mathcal{X}]$ suitably chosen so as to obtain $\mathcal{M}_g < \mathcal{M}_f$, cf. Proposition 2.5. The iteration continues until no monomial occurring in the result is divisible by the leading term of a polynomial from B. In other words, we are describing *StronglyReduce*(B, f). In order for the result to be in U, the leading monomial of any element of (B) must be a multiple of the leading monomial of at least one member of B. But that is a nontrivial condition! In fact, by Theorem 3.10, it is equivalent to B being a Gröbner basis. We summarize:

Proposition 4.4. *Suppose B is a Gröbner basis of $K[\mathcal{X}]$. If U is the linear span of all standard monomials, then*

$$K[\mathcal{X}] = U \oplus (B).$$

Moreover, StronglyReduce(B, f) projects f into U.

Thus, the algebraic operations of a K-algebra $A = K[\mathcal{X}]/(B)$ are effective: First compute $C = \mathit{GroebnerBasis}(B)$. The elements of A are the residue classes $f+(B)$, which are represented by f. Then the equality between $f+(B)$ and $g+(B)$ is tested by equating the two polynomials $\mathit{StronglyReduce}(C, f)$ and $\mathit{StronglyReduce}(C, g)$. Now multiplication on A works as follows: do the arithmetic in $K[\mathcal{X}]$ and apply $\mathit{StronglyReduce}(C, \cdot)$ to the result. Note that *StronglyReduce* is not needed for subtraction and addition.

Example 4.5. For B as in Example 3.1, we have seen that $C = \{X-Y, Y^3-1\}$ is a Gröbner basis (if $X > Y$). Thus $A = K[\mathcal{X}]/(B)$ can be thought of as the linear subspace U of $K[\mathcal{X}]$ spanned by $\{1, Y, Y^2\}$, and squaring Y^2 gives $\mathit{StronglyReduce}(C, Y^4) = Y \in U$.

Exercise 4.6. Suppose B is a Gröbner basis in $K[\mathcal{X}]$ with respect to a fixed reduction order. Prove that $\mathit{StronglyReduce}(B, f) = \mathit{StronglyReduce}(B, g)$ holds if and only $f + (B) = g + (B)$.

Remark 4.7. The set $K[\mathcal{X}]$ is used to represent polynomials, not only for $K[\mathcal{X}]$ itself but also for the quotient ring $K[\mathcal{X}]/I$, where I is an ideal of $K[\mathcal{X}]$. We have seen that, by use of *StronglyReduce* and *Reduce*, these representatives can be controlled. The ultimate control is the existence of a unique element in $K[\mathcal{X}]$ for a whole class in $K[\mathcal{X}]/I$. Such an element is often referred to as a *normal form*, especially when it can be computed starting from an arbitrary representative by means of rewritings. This is achieved by *StronglyReduce* with respect to a Gröbner basis. Thus, we shall also refer to $\mathit{StronglyReduce}(B, f)$ as a *normal form of f* if B is a Gröbner basis.

Example 4.8. A *StronglyReduce* algorithm for the Grassmann variety of lines on a vector space V was known before Gröbner bases. Recall that $\bigwedge^2 V$, the exterior square of V, is the vector space that can be obtained as the -1-eigenspace of $V \otimes V$ with respect to the involution $\sigma : V \otimes V \to V \otimes V$ interchanging the two factors; that is, $\sigma(x \otimes y) = y \otimes x$ for $x, y \in V$. (K is assumed to have characteristic distinct from 2.) It has dimension $\binom{n}{2}$ and coordinate functions X_{ij} for $1 \le i < j \le n$, where $n = \dim V$, which can be extended by the rules $X_{ii} = 0$ and $X_{ji} = X_{ij}$. Recall that 1-dimensional linear subspaces spanned by pure wedges $x \wedge y = x \otimes y - y \otimes x$ in $\bigwedge^2 V$ represent 2-dimensional subspaces of V (namely, the linear span of x and y). The set of all single wedges is actually an algebraic variety: it is the zeroset of the quadratic polynomials

$$P_{ij;kl} = X_{ij}X_{kl} + X_{jk}X_{il} + X_{ki}X_{jl}. \tag{4.1}$$

Using these equations, it is easily derived that modulo the ideal generated by the $P_{ij;kl}$, one can always rewrite a monomial to a linear combination of monomials $X_{ij}X_{kl}$, each of which satisfies $i < j$, $k < l$, and $i \leq k$, $j \leq l$. This observation can be used to prove that the polynomials (4.1) form a Gröbner basis and that those just mentioned form the set of all standard monomials. Thus, if we denote their linear span by U, we have

$$K[\{X_{ij} \mid 1 \leq i < j \leq n\}] = U \oplus (\{P_{ij;kl} \mid i < j, k < l\}).$$

In particular, the quadratic polynomials $P_{ij;kl}$ form a Gröbner basis with respect to any total degree lexicographic order extending the partial order on the variables given by

$$X_{ij} \leq X_{kl} \Leftrightarrow i \leq k \text{ and } j \leq l \quad (\text{where } i < j,\ \ k < l).$$

Exercise 4.9. Verify that, if $\dim V = 4$, the Grassmann variety of lines is just a (single) quadric. Consider the Grassmann ring for $\dim V = 5$, that is, $K[X_{12}, \ldots, X_{45}]/(\{P_{ij;kl}\})$. Write out $X_{15}X_{23}X_{34}$ as a sum of standard monomials.

5. Solving Polynomial Equations

We now address Problem 2: Suppose we have a vector $\mathbf{a} \in K[\mathcal{X}]^\ell$. We want to describe the subset of K^n consisting of all solutions $\mathbf{x} \in K^n$ of the system of equations

$$a_1(\mathbf{x}) = a_2(\mathbf{x}) = \cdots = a_\ell(\mathbf{x}) = 0.$$

We shall denote this subset of K^n by $\mathcal{Z}(\mathbf{a})$, or $\mathcal{Z}(A)$ if $A = \{a_1, \ldots, a_\ell\}$.

Before dealing with the general case, note that, at first year university level (or earlier), we were taught how to deal with such a system if $n = 1$ (the univariate case) or if all a_i have degree 1 (the linear case). Let us briefly recall these two particular cases.

The Univariate Case

If $\mathcal{X} = \{X_1\} = \{X\}$, we can use the Euclidean Algorithm to reduce the equations $a_1(x) = \cdots = a_\ell(x) = 0$ to the single one $f(x) = 0$, where $f \in K[\mathcal{X}]$ is the gcd of $\{a_1, \ldots, a_\ell\}$. Once the system is reduced to a single univariate polynomial equation, all we can do further is factor the polynomial into irreducible ones. Each irreducible factor will then have as solutions formal elements in the extension field that it defines. In particular, solutions in K correspond to the linear factors of f.

Example 5.1. Consider the polynomials

$$\begin{aligned} a_1 &= X^{16} - 7\,X^{12} + 18\,X^8 - 21\,X^4 + 10, \\ a_2 &= X^{12} - 6\,X^8 - X^5 + 11\,X^4 + 2\,X - 6. \end{aligned}$$

The gcd of these two polynomials is $a_0 = X^4 - 2$. This is an irreducible polynomial. In particular, there are no rational solutions. Each solution can be described as an element x of an extension field isomorphic to $\mathbb{Q}[X]/(X^4 - 2)$. Inside the latter field we can describe one solution as $x = X + (a_0)$. If we want to describe all solutions in these terms, we need the so-called splitting field of a_0. Here, clearly $-x$ is another solution. Further solutions are zeros of the quotient $c_0 = X^2 + x^2$ of $X^4 - 2$ by $(X - x)(X + x)$. Since this quotient is irreducible over $\mathbb{Q}(x)$, the quadratic extension $\mathbb{Q}(x)[X]/(c_0)$ is needed for a description of the two remaining solutions. Within that extension, the element $y = X + c_0\mathbb{Q}(x)[X]$ is a solution, as well as $-y$.

Because of the particular form of a_0, we can describe x by the more familiar notation $x = \sqrt[4]{2}$. The additional information in this notation is that it refers to the biggest real root among all four. Similarly, y can then be identified with $ix = i\sqrt[4]{2}$. To see this, note that $z = y/x$ satisfies $z^2 = -1$ (as $y^2 = -x^2$ in $\mathbb{Q}(x)[X]/(c_0)$), so that it can indeed be identified with i.

In most cases (as explained by Galois theory) it is not possible to find such an explicit description of a root in K. For instance, for a zero of the irreducible polynomial $X^5 + X + 1$, one cannot do much better than produce the tautology $x =$`RootOf`$(X^5 + X + 1, X)$. If K is an algebraic number field, one way of distinguishing the five roots is by describing their embeddings in the complex plane. For instance, one might refer to the biggest (or one but biggest) real solution, or the complex solution with greatest real part, etc.

In conclusion, solving a system of univariate equations comes down to solving a single univariate equation, namely the gcd of the original polynomials. In order to find explicit descriptions of solutions, factorization of univariate polynomials over field extensions is needed. See Chapter 4 for a treatment of this topic. Suffice it to say here that, in general, it is not easy to find a factorization over an extension field.

The Linear Homogeneous Case

Suppose now that all a_i are linear in the variables $\mathcal{X} = \{X_1, \ldots, X_n\}$. Then the usual solution method is Gauss elimination. We order the variables, say $X_1 < \cdots < X_n$. Suppose, without loss of generality, that the monomial X_n occurs in a_ℓ (that is, it occurs in a_ℓ as the monomial of a term with nonzero coefficient). Denote by a^{ij} the coefficient of X_j in a_i. Thus, $a^{\ell n} \neq 0$. A new system of $\ell - 1$ linear equations is formed by replacing a_j by $a'_j = a^{\ell n}a_j - a^{jn}a_\ell$ for $j \neq \ell$, so X_n cancels out. The equation $a_\ell = 0$ is kept apart for later use, namely to express the coordinate x_n of solutions $\mathbf{x}$ in terms of other coordinates. The system $a'_j = 0$ may collapse as some of the a'_j could be zero.

Apart from this phenomenon, there are no complications: one continues with the a'_j ($j \neq \ell$), eliminating the next variable down from X_n that occurs in one of these. If, at some stage, a variable X_j has not been set aside as yet and does not occur in the resulting system, then x_j will be a free parameter for the solution x. The final result has a set of free parameters, and —due to the equations kept apart— a way of expressing each of the remaining coordinates x_j in terms of these parameters.

Example 5.2. Let $K = \mathbb{Q}$ and consider the polynomials

$$\begin{aligned} a_1 &= 8X_1 - 18X_2 - 9X_3 - 6X_4, \\ a_2 &= 4X_1 - 10X_2 - 6X_3 - 4X_4, \\ a_3 &= 2X_1 - 6X_2 - 3X_3 - 2X_4. \end{aligned}$$

The corresponding system of equations in two variables, derived according to the above procedure, with $\ell = 3$ and $n = 4$, is

$$\begin{aligned} a'_1 &= a_1 - 3a_3 = 2X_1, \\ a'_2 &= a_2 - 2a_3 = 2X_2. \end{aligned}$$

As X_3 does not occur in a'_1 and a'_2, the variable X_3 can be assigned any value in $\mathbb{Q}$. The two remaining equations are already in upper triangular shape; they express $x_1 = 0$ and $x_2 = 0$, respectively. The equation set apart (coming from $a_3 = 0$) then expresses x_4 in terms of x_3, and so all solutions look like

$$(0, 0, x_3, -3x_3/2) \text{ with } x_3 \in \mathbb{Q}.$$

The Role of the Buchberger Algorithm

The general solving strategy is a blend of the two methods encountered in the above special cases.

It is important to observe that the set of equations $a_1 = \cdots = a_\ell = 0$ is equivalent to any other set $b_1 = \cdots = b_k = 0$ for which $\{b_1, \ldots, b_k\}K[\mathcal{X}] = \{a_1, \ldots, a_\ell\}K[\mathcal{X}]$. The idea behind the solving procedure is to replace the set $\{a_1, \ldots, a_\ell\}$ by a more suitable ('upper triangular') set: a Gröbner basis with respect to a lexicographic order.

Example 5.3. In the univariate case, every Gröbner basis of $\{a_1, \ldots, a_\ell\}$ contains $f = \gcd(a_1, \ldots, a_\ell)$, and the singleton $\{f\}$ is a Gröbner basis of $(\{a_1, \ldots, a_\ell\})$. Thus, Gröbner bases achieve the same reduction as described above.

Before going over to more variables, we fix $<$ to be the lexicographic order on $\mathcal{M}$ with $X_1 < X_2 < \cdots < X_n$.

Example 5.4. Put $A = \{a_1, \ldots, a_\ell\}$. In the linear case, with X_n occurring in a_ℓ, we can perform *GroebnerBasis*(**a**) in such a way that we compute *Reduce*$(A, S(a_j, a_\ell))$ first, for all $j < \ell$. Note that the S-polynomial involved coincides with the reduction from a_j to a_j' described in the treatment of the linear case above. The reduction takes care of further elimination of variables in a_j' as is done later in the Gauss elimination. In our short description of *GroebnerBasis* we did not throw out elements from the basis at hand. But here it is clear that $A \cup \{a_j'\} \setminus \{a_j\}$ and A generate the same ideal and that the leading terms of $A \cup \{a_j'\} \setminus \{a_j\}$ and of $A \cup \{a_j'\}$ also generate the same ideal, so we may remove a_j in the Gröbner basis computation once a_j' has been adjoined. Continuing this way, we do not increase the number of equations. After all reductions of $S(a_j, a_\ell)$ have been dealt with, the equation a_ℓ will have stayed apart, as its leading term X_n does not occur in any of the other equations. Thus the same pattern emerges as with Gauss elimination.

Solving Polynomial Equations

Of great importance for solving equations is the following result.

Proposition 5.5 (Elimination of Variables). *If B is a Gröbner basis with respect to the lexicographic order $<$ with $X_1 < \cdots < X_n$, and if $i < n$, then*

$$K[X_1, \ldots, X_i] \cap (B) = (K[X_1, \ldots, X_i] \cap B) K[X_1, \ldots, X_i].$$

Proof. The inclusion $\supseteq$ is obvious. Let $f \in K[X_1, \ldots, X_i] \cap (B)$. Since $f \in (B)$ and B is a Gröbner basis, there are h_b for $b \in B$ with $f = \sum_{b \in B} h_b b$ and $\mathrm{lm}(h_b b) \leq \mathrm{lm}(f)$. But $f \in K[X_1, \ldots, X_i]$, so $\mathrm{lm}(f) < X_{i+1}$. Hence also $\mathrm{lm}(h_b b) < X_{i+1}$, showing that $\mathrm{lm}(h_b b)$ lies in $K[X_1, \ldots, X_i]$ for each $b \in B$. In particular, $b \in K[X_1, \ldots, X_i] \cap B$ whenever $h_b \neq 0$. Since $h_b \in K[X_1, \ldots, X_i]$, this establishes $f = \sum_b h_b b \in (K[X_1, \ldots, X_i] \cap B) K[X_1, \ldots, X_i]$.

Exercise 5.6. An order with $X_1 < \cdots < X_n$ is called an *elimination order* if $X_i^j < X_{i+1}$ for any j and $i = 1, \ldots, n-1$. Show that Proposition 5.5 is also valid for any elimination order instead of the lexicographic order.

The proposition guarantees that, whatever solutions $\mathbf{x} \in K^n$ to $b(\mathbf{x}) = 0$ for $b \in B$ there may be, their first coordinates x_1 will satisfy the gcd of all equations in (B) without variables $X_2, \ldots, X_n$. More precisely:

Proposition 5.7. *Suppose B is a Gröbner basis with respect to the lexicographic order with $X_1 < \cdots < X_n$. Let $(x_1, \ldots, x_\ell) \in K^\ell$ for some $\ell \leq n$. Then, for $1 \leq \ell \leq n$, we have*

1. *if $(x_1, \ldots, x_n) \in \mathcal{Z}(B)$ then $(x_1, \ldots, x_\ell) \in \mathcal{Z}(B \cap K[X_1, \ldots, X_\ell])$;*
2. *if $C = \{f(x_1, \ldots, x_\ell, X_{\ell+1}, \ldots, X_n) \mid f \in B\}$ then*

$$\mathcal{Z}(C) = \{(x_{\ell+1}, \ldots, x_n) \in K^{n-\ell} \mid (x_1, x_2, \ldots, x_n) \in \mathcal{Z}(B)\};$$

3. the set C of 2. is a Gröbner basis in $K[X_{\ell+1},\ldots,X_n]$.

Proof. 1. If $f \in K[X_1,\ldots,X_\ell]$ then, viewing f as a polynomial in the variables $X_1,\ldots,X_n$ which is constant in $X_{\ell+1},\ldots,X_n$, we have $f(x_1,\ldots,x_\ell) = f(x_1,\ldots,x_\ell,x_{\ell+1},\ldots,x_n)$. If, moreover, $f \in B$, then $f(x_1,\ldots,x_n) = 0$, and so $f(x_1,\ldots,x_\ell) = 0$, proving $(x_1,\ldots,x_\ell) \in \mathcal{Z}(B \cap K[X_1,\ldots,X_\ell])$.

2. Immediate from the definition of C.

3. First observe that if $X_1^{a_1}\cdots X_n^{a_n} \leq X_1^{b_1}\cdots X_n^{b_n}$, then $X_{\ell+1}^{a_{\ell+1}}\cdots X_n^{a_n} \leq X_{\ell+1}^{b_{\ell+1}}\cdots X_n^{b_n}$. In particular, if $f,g \in K[\mathcal{X}]$ satisfy $\mathrm{lm}(f) \leq \mathrm{lm}(g)$, then $\mathrm{lm}(f(x_1,\ldots,x_\ell,X_{\ell+1},\ldots,X_n)) \leq \mathrm{lm}(g(x_1,\ldots,x_\ell,X_{\ell+1},\ldots,X_n))$.

Let $g \in CK[X_{\ell+1},\ldots,X_n]$. Then there are $h_b \in K[X_{\ell+1},\ldots,X_n]$ for $b \in B$ such that

$$g = \sum_{b\in B} h_b b(x_1,\ldots,x_\ell,X_{\ell+1},\ldots,X_n).$$

Consider $\tilde{g} = \sum_{b\in B} h_b b(X_1,\ldots,X_\ell,X_{\ell+1},\ldots,X_n) \in K[\mathcal{X}]$. Since $\tilde{g} \in (B)$ and B is a Gröbner basis, there are $\tilde{h}_b \in K[\mathcal{X}]$ with $\tilde{g} = \sum_{b\in B} \tilde{h}_b b$ and $\mathrm{lm}(\tilde{h}_b b) \leq \mathrm{lm}(\tilde{g})$ for all $b \in B$. The leading monomial of $\tilde{g}$ has the form $X_1^{a_1}\cdots X_n^{a_n}$ for certain $a_1,\ldots,a_n \in \mathbb{N}$, with $\mathrm{lm}(g) = X_{\ell+1}^{a_{\ell+1}}\cdots X_n^{a_n}$. By the observation at the beginning of the proof,

$$\begin{aligned}&\mathrm{lm}(\tilde{h}_b(x_1,\ldots,x_{\ell-1},X_\ell,\ldots,X_n)b(x_1,\ldots,x_{\ell-1},X_\ell,\ldots,X_n)) \leq\\ &\leq \mathrm{lm}(\tilde{g}(x_1,\ldots,x_{\ell-1},X_\ell,\ldots,X_n)) = \mathrm{lm}(g).\end{aligned}$$

with equality for at least one $b \in B$. For such a polynomial b we have, since $b(x_1,\ldots,x_{\ell-1},X_\ell,\ldots,X_n) \in C$,

$$\begin{aligned}\mathrm{lm}(g) &\in \mathrm{lm}(b(x_1,\ldots,x_{\ell-1},X_\ell,\ldots,X_n))K[X_{\ell+1},\ldots,X_n] \subseteq\\ &\subseteq \{\mathrm{lm}(c) \mid c \in C\}K[X_{\ell+1},\ldots,X_n],\end{aligned}$$

whence

$$\{\mathrm{lm}(g) \mid g \in (C)\}K[X_{\ell+1},\ldots,X_n] = \{\mathrm{lm}(c) \mid c \in C\}K[X_{\ell+1},\ldots,X_n].$$

By Theorem 3.10, C is a Gröbner basis in $K[X_{\ell+1},\ldots,X_n]$. This proves Part 3.

As a consequence of this proposition, we can reduce the solution of $b_1 = \cdots = b_n = 0$ to solving polynomial equations in fewer variables.

Example 5.8. Let us solve $b_1 = b_2 = 0$ with $B = \{b_1, b_2\}$ as in Example 3.1. In the Gröbner basis of B there is a single polynomial in the single variable Y (namely b_4, see Example 3.14), which tells us that $Y = \sqrt[3]{1}$. The next equation up (namely $b_3 = 0$) tells us $X = Y$. This completely determines the solution set:

$$\mathcal{Z}(B) = \{(1,1),(e^{2\pi i/3},e^{2\pi i/3}),(e^{4\pi i/3},e^{4\pi i/3})\}.$$

Example 5.9. Consider $\mathbf{b} = (X^2 + Y^2 + Z^2, XYZ) \in K[X,Y,Z]^2$, where K is an effective subfield of $\mathbb{C}$. The reduced Gröbner basis with respect to the lexicographic order with $X > Y > Z$ can be easily computed by use of a computer algebra software package. The result is $B = \{b_1, b_2, b_3\}$ where $b_3 = Y^3Z + Z^3Y$. Since B has no univariate polynomial in Z, we have $\mathcal{Z}(B \cap K[Z]) = \mathcal{Z}(\emptyset) = K^3$. There is one element in B that is a polynomial in the variables Y and Z only, viz., b_3. Solving $b_3 = 0$, we find that

$$\mathcal{Z}(B \cap K[Y,Z]) = \mathcal{Z}(b_3) = \{(x,y,z) \mid y = 0 \vee z = 0 \vee y = iz \vee y = -iz\}$$

provided $i = \sqrt{-1} \in K$. Thus, it is the union of the four lines $\{Y = 0\}$, $\{Z = 0\}$, $\{(y, iy) \mid y \in K\}$, $\{(y, -iy) \mid y \in K\}$ of K^2. This distinction of cases propagates as we add the variable X, a phenomenon typical of what can be expected in general. In this case, the result is a union of lines in K^3.

More about solving polynomial systems can be found in Chapter 2.

Exercise 5.10. Recall that the minimal polynomial of an algebraic number $x \in \overline{K}$ is the polynomial of minimal positive degree in $K[X]$ of which x is a zero. Let f be as above. Show that the following method finds the minimal polynomial of $g + (f) \in K[X]/(f)$, for $g \in K[X]$: Compute the Gröbner basis B of $(g - Y, f) \in K[X,Y]^2$ with respect to the lexicographic order with $X > Y$. Then $B \cap K[Y]$ is generated by the minimal polynomial (with variable Y) for g.

Exercise 5.11. The previous exercise may be seen as a special case of the following more general determination of the image and kernel of a morphism. Let I be an ideal of $K[\mathcal{X}]$ generated by a finite set B and let $\phi : K[\mathcal{Y}] \to K[\mathcal{X}]/I$ be a morphism of K-algebras. Prove the following assertions.

1. ϕ is determined by polynomials $F_j \in K[\mathcal{X}]$ such that $\phi(Y_j) = F_j + I$ $(j = 1, \ldots, m)$, where $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$.
2. Let J be the kernel of the morphism
$$\phi^* : K[\mathcal{X} \cup \mathcal{Y}] \to K[\mathcal{X} \cup \mathcal{Y}]/(B)$$
given by $\phi^*(Y_j) = F_j + (B)$ and $\phi^*(X_i) = X_i + (B)$. Then J is generated by $Y_j - F_j$, $(j = 1, \ldots, m)$ and B. Hint: for $g \in \ker \phi^*$, write $g = g - g(X_1, \ldots, X_n, F_1, \ldots, F_m)$ as a linear combination of 'binomials' $Y^a - F^a$ $(a \in \mathbb{N}^m)$ and show by induction on a that such binomials are in the ideal generated by $Y_j - F_j$.
3. $\ker \phi = J \cap K[\mathcal{Y}]$.
4. $\ker \phi$ can be computed by means of a Gröbner basis algorithm applied to the above generating set for J with respect to an elimination order (cf. Proposition 5.5) satisfying $Y_j^c < X_i$ for all $c, i, j \in \mathbb{N}$.
5. $f \in K[\mathcal{X}]$ satisfies
$$f + I \in \operatorname{im} \phi \quad \Leftrightarrow \quad \mathit{StronglyReduce}(G, f) \in K[\mathcal{Y}],$$
where G is a Gröbner basis for J as suggested above.

6. Effectiveness of Polynomial Rings

In this section we are concerned with Problem 1: effectiveness of $K[\mathcal{X}]$. As we have seen, this amounts to describing all solutions $\mathbf{x} = (x_1, \ldots, x_\ell) \in K[\mathcal{X}]^\ell$ of the equation

$$\sum_{1\leq i\leq \ell} a_i x_i \quad = \quad b, \tag{6.1}$$

for $\mathbf{a} = (a_1, \ldots, a_\ell) \in K[\mathcal{X}]^\ell$ and $b \in K[\mathcal{X}]$. We shall also use vector notation and write $\mathbf{a}\mathbf{x}^\top = b$ for this equation.

Determining whether there is a solution to (6.1) is the same as the ideal membership problem for the ideal $I = \{a_1, \ldots, a_\ell\}K[\mathcal{X}]$ and the polynomial b, that is, $b \in I$ if and only if Equation (6.1) has a solution. But, by the definition of Gröbner basis and Corollary 3.12, b is a member of I if and only if $\mathit{Reduce}(\mathit{GroebnerBasis}(\{a_1, \ldots, a_\ell\}), b) = 0$, and so this can be decided effectively. This observation already indicates that solving (6.1) is easier if $A = \{a_1, \ldots, a_\ell\}$ is a Gröbner basis.

Suppose we know $\mathit{Reduce}(B, f) = 0$. Then $f \in (B)$, and the more elaborate version below of the Reduction Algorithm gives us a vector $\mathbf{c} \in K[\mathcal{X}]^\ell$ of coefficients for which the membership $f \in (B)$ is realised, that is, such that $f = \mathbf{b}\mathbf{c}^\top$, where $\mathbf{b} \in K[\mathcal{X}]^\ell$ is such that $B = \{b_1, \ldots, b_\ell\}$.

```
$\mathit{ExtendedReduce}(\mathbf{b}, f)$ =
        $J := \{i \in \{1, \ldots, \ell\} \mid \mathrm{lm}(b_i)|\mathrm{lm}(f)\}$;
        if $J \neq \emptyset$
        then choose $j \in J$;
          $(\mathbf{c}, g) := \mathit{ExtendedReduce}(\mathbf{b}, f - (\mathrm{lt}(f)/\mathrm{lt}(b_j))b_j)$;
          return $(\mathbf{c} + (\mathrm{lt}(f)/\mathrm{lt}(b_j))\epsilon_j, g)$
        else return $(0, f)$
        fi.
```

Here ϵ_j $(1 \leq j \leq \ell)$ denotes the standard basis of the $K[\mathcal{X}]$-module $K[\mathcal{X}]^\ell$. We describe the functionality of the algorithm in the following

Proposition 6.1. *For* $\mathbf{b} \in K[\mathcal{X}]^\ell$ *and* $f \in K[\mathcal{X}]$, *the algorithm Extended-Reduce finds a pair* $(\mathbf{c}, g)$ *consisting of a vector* $\mathbf{c} \in K[\mathcal{X}]^\ell$ *and a reduced form* $g \in K[\mathcal{X}]$ *of* f *such that* $f - g = \mathbf{b}\mathbf{c}^\top$ *and* $\mathrm{lm}(b_i c_i) \leq \mathrm{lm}(f)$ *for* $i = 1, \ldots, \ell$.

Let $\mathbf{a} \in K[\mathcal{X}]^\ell$. If we have found one solution $\mathbf{d}$ of the equation $\mathbf{a}\mathbf{x}^\top = b$, then any solution is of the form $\mathbf{d}+\mathbf{y}$ where $\mathbf{y}$ is a solution of the homogeneous equation $\mathbf{a}\mathbf{y}^\top = 0$.

If $\{a_1, \ldots, a_\ell\}$ is a Gröbner basis, then *ExtendedReduce*$(\mathbf{a}, b)$ produces a particular solution to $\mathbf{a}\mathbf{x}^\top = b$. Therefore, the main problem is the homogeneous equation $\mathbf{a}\mathbf{x}^\top = 0$. Note that its solutions form a $K[\mathcal{X}]$-submodule of $K[\mathcal{X}]^\ell$; in fact they form the kernel of the $K[\mathcal{X}]$-linear map $\phi_\mathbf{a} : K[\mathcal{X}]^\ell \to K[\mathcal{X}]$ given by $\phi_\mathbf{a}(\mathbf{x}) = \mathbf{a}\mathbf{x}^\top$. Thus, solving the homogeneous equation can be conveniently rephrased as the problem of finding a set of generators of $\ker \phi_\mathbf{a}$.

Definition 6.2. Elements of $\ker \phi_\mathbf{a}$ describe relations among the individual components of $\mathbf{a}$; such relations are called *syzygies*.

We shall find a finite spanning set for the syzygies of $\mathbf{a}$.

Syzygies for Vectors of Terms

First we treat the case where $\mathbf{a}$ is a *vector of terms*, that is, each component a_j is a term. Note that this solves the problem of finding a spanning set for the sygyzies corresponding to a monomial ideal, that is, an ideal generated by a set of monomials. Theorem 6.9 below shows how to deal with the general case.

If $\mathbf{a}$ is fixed and $\mathbf{v}$ has to be chosen so as to cancel out a_i and a_j, we can take $\mathbf{v}$ with nonzero components at the indices i and j only, leaving $a_i v_i + a_j v_j$ to be made zero. This can be done similar to the construction of the S-polynomial.

For $1 \le i \le j \le \ell$ with $a_i, a_j \ne 0$, we define

$$\begin{aligned} m_\mathbf{a}^{ij} &= \operatorname{lcm}(\operatorname{lm}(a_i), \operatorname{lm}(a_j)), \quad \text{and} && (6.2)\\ \mathbf{v}_\mathbf{a}^{ij} &= (m_\mathbf{a}^{ij}/\operatorname{lt}(a_i))\epsilon_i - (m_\mathbf{a}^{ij}/\operatorname{lt}(a_j))\epsilon_j. && (6.3)\end{aligned}$$

Lemma 6.3. *Let* $\mathbf{a} \in K[\mathcal{X}]^\ell$ *be a nonzero vector of terms with* $a_k \ne 0$ *for all* k. *Then the elements* $\mathbf{v}_\mathbf{a}^{ij}$ $(1 \le i < j \le \ell)$ *of* $K[\mathcal{X}]^\ell$ *generate* $\ker \phi_\mathbf{a}$ *as a* $K[\mathcal{X}]$*-submodule. Moreover, for each* $w \in \ker \phi_\mathbf{a}$, *there are* $f_{ij} \in K[\mathcal{X}]$ *with* $\operatorname{lm}(f_{ij} m_\mathbf{a}^{ij}) \le \operatorname{lm}(a_i w_i)$ *such that*

$$\mathbf{w} = \sum_{1 \le i < j \le \ell} f_{ij} \mathbf{v}_\mathbf{a}^{ij}.$$

Proof. As $\mathbf{a}$ is a vector of terms, we have $a_k = \operatorname{lt}(a_k)$ for each k. Consequently, for all i, j,

$$\mathbf{a}(\mathbf{v}_\mathbf{a}^{ij})^\top = a_i m_\mathbf{a}^{ij}/\operatorname{lt}(a_i) - a_j m_\mathbf{a}^{ij}/\operatorname{lt}(a_j) = 0,$$

so that the $K[\mathcal{X}]$-submodule generated by the $\mathbf{v}_\mathbf{a}^{ij}$ is contained in $\ker \phi_\mathbf{a}$.

As for the converse, suppose $\mathbf{w} \in \ker \phi_\mathbf{a}$. If $\mathbf{w} = 0$, or there is just one coordinate in which $\mathbf{w}$ is nonzero, there is nothing to prove. Therefore, we assume that there are two distinct indices, say i, j, with $i < j$, such that $w_i, w_j \ne 0$ and $\operatorname{lm}(a_i w_i) = \operatorname{lm}(a_j w_j)$. In particular, there are a term u and a coefficient $c \in K$ with $\operatorname{lt}(a_i w_i) = u m_\mathbf{a}^{ij} = c \operatorname{lm}(a_j w_j)$. Now $\mathbf{w}' = \mathbf{w} - u\mathbf{v}_\mathbf{a}^{ij}$ has

$$\begin{aligned} w_i' &= w_i - um_{\mathbf{a}}^{ij}/\mathrm{lt}(a_i) = w_i - \mathrm{lt}(w_i), \\ w_j' &= w_j + um_{\mathbf{a}}^{ij}/\mathrm{lt}(a_j) = w_j + c\,\mathrm{lm}(w_j)/\mathrm{lc}(a_j), \text{ and} \\ w_k' &= w_k \text{ for } k \neq i, j. \end{aligned}$$

Note that $w \in \ker \phi_{\mathbf{a}}$ implies that $w' \in \ker \phi_{\mathbf{a}}$ by the first part of this proof. Using Part 1 of Exercise 2.6 applied with $B = \{1, \ldots, \ell\}$, we see that the map $B \to \mathcal{M}$, $k \mapsto \mathrm{lm}(w_k')$ is smaller than $k \mapsto \mathrm{lm}(w_k)$. Thus repeatedly replacing $\mathbf{w}$ by $\mathbf{w}'$ as above, we find a descending chain which must stop after finitely many steps. But it only stops if $\mathbf{w} = 0$. Hence, $\mathbf{w} \in \ker \phi_{\mathbf{a}}$.

The last assertion of the lemma follows from an estimate of u in terms of the coordinates of $\mathbf{w}'$ and $\mathbf{w}$ above. Indeed, f_{ij} is built up of terms such as u and $\mathrm{lm}(um_{\mathbf{a}}^{ij}) \leq \mathrm{lm}(w_i)\mathrm{lm}(a_i)$.

Exercise 6.4. For (X^2Y, XY^2, X, Y^3) in $\mathbb{Q}[X,Y]^4$, find a finite spanning set of syzygies.

So far, we have provided a solution to Equation (6.1) for the case where $\mathbf{a}$ consists of terms and $b = 0$.

Syzygies for Gröbner Bases

Still we are not ready for the general case. We pass to a vector $\mathbf{a} \in K[\mathcal{X}]^\ell$ such that $A = \{a_1, \ldots, a_\ell\}$ is a Gröbner basis with $a_i \neq 0$ for all i. Then 0 is the reduced form of $S(a_i, a_j)$ modulo A for any i, j. In view of (3.1), this means that there is a vector $\mathbf{h}^{ij} \in K[\mathcal{X}]^\ell$ with

$$S(a_i, a_j) \quad = \quad \mathbf{a}(\mathbf{h}^{ij})^\top \text{ and } \mathrm{lm}(a_k h_k^{ij}) < m_{\mathrm{lt}(\mathbf{a})}^{ij} \text{ for all } k, \qquad (6.4)$$

where $\mathrm{lt}(\mathbf{a})$ stands for the vector of leading terms of $\mathbf{a}$. The vector $\mathbf{h}^{ij}$ can be found by means of the ExtendedReduce algorithm; it is the first component of the output of *ExtendedReduce*$(\mathbf{a}, S(a_i, a_j))$.

But also

$$S(a_i, a_j) = (m_{\mathrm{lt}(\mathbf{a})}^{ij}/\mathrm{lt}(a_i))a_i - (m_{\mathrm{lt}(\mathbf{a})}^{ij}/\mathrm{lt}(a_j))a_j = \mathbf{a}(\mathbf{v}_{\mathrm{lt}(\mathbf{a})}^{ij})^\top.$$

Thus, $\mathbf{v}_{\mathrm{lt}(\mathbf{a})}^{ij} - \mathbf{h}^{ij} \in \ker \phi_{\mathbf{a}}$.

Theorem 6.5. *Suppose that* $\mathbf{a} \in K[\mathcal{X}]^\ell$ *is such that* $\{a_1, \ldots, a_\ell\}$ *is a Gröbner basis and all* $a_i \neq 0$. *Then the submodule* $\ker \phi_{\mathbf{a}}$ *of* $K[\mathcal{X}]^\ell$ *is generated by all*

$$\mathbf{v}_{\mathrm{lt}(\mathbf{a})}^{ij} - \mathbf{h}^{ij} \text{ with } 1 \leq i < j \leq \ell;$$

here $\mathbf{v}_{\mathrm{lt}(\mathbf{a})}^{ij}$ *is defined as in (6.3) and* $\mathbf{h}^{ij}$ *as in (6.4).*

Proof. Suppose $\mathbf{v} \in \ker \phi_{\mathbf{a}}$. Then $\mathrm{lt}(\mathbf{v}) \in \ker \phi_{\mathrm{lt}(\mathbf{a})}$ and so, by Lemma 6.3, there are $f_{ij} \in K[\mathcal{X}]$ with

$$\mathrm{lt}(\mathbf{v}) = \sum_{i,j} f_{ij} \mathbf{v}^{ij}_{\mathrm{lt}(\mathbf{a})} \quad \text{and} \quad \mathrm{lm}(f_{ij} m^{ij}_{\mathrm{lt}(\mathbf{a})}) \leq \mathrm{lm}(a_i v_i).$$

Now consider

$$\mathbf{v}' = \mathbf{v} - \sum_{i,j} f_{ij} (\mathbf{v}^{ij}_{\mathrm{lt}(\mathbf{a})} - \mathbf{h}^{ij}),$$

where $\mathbf{h}^{ij}$ is as in (6.4). Since $\mathbf{v}^{ij}_{\mathrm{lt}(\mathbf{a})} - \mathbf{h}^{ij} \in \ker \phi_{\mathbf{a}}$, the vector $\mathbf{v}'$ also belongs to $\ker \phi_{\mathbf{a}}$. By the inequality of (6.4) we find, for each $s \in \{1, \ldots, \ell\}$,

$$\mathrm{lm}(a_s f_{ij} h^{ij}_s) < \mathrm{lm}(f_{ij} m^{ij}_{\mathrm{lt}(\mathbf{a})}) \leq \mathrm{lm}(a_i v_i) \leq \max\{\mathrm{lm}(a_k v_k) \mid 1 \leq k \leq \ell\}.$$

Hence $\max\{\mathrm{lm}(a_k v'_k) \mid 1 \leq k \leq \ell\} < \max\{\mathrm{lm}(a_k v_k) \mid 1 \leq k \leq \ell\}$. Therefore, we can apply induction to conclude that every vector of $\ker \phi_{\mathbf{a}}$ is a $K[\mathcal{X}]$-linear combination of the $\mathbf{v}^{ij}_{\mathrm{lt}(\mathbf{a})} - \mathbf{h}^{ij}$.

By now we can solve the equation at the beginning of the section.

Corollary 6.6. *A complete solution to Equation (6.1) in case* $\mathbf{a}$ *corresponds to a Gröbner basis is obtained as follows: Compute*

$$(\mathbf{d}, g) := \mathit{ExtendedReduce}(\mathbf{a}, b).$$

If $g \neq 0$, then there are no solutions. Otherwise, $\mathbf{d}$ *is a particular solution and an arbitrary solution is of the form*

$$\mathbf{d} + \sum_{i,j} f_{ij} (\mathbf{v}^{ij}_{\mathrm{lt}(\mathbf{a})} - \mathbf{h}^{ij}) \quad \text{with } f_{ij} \in K[\mathcal{X}],$$

for $\mathbf{h}$ *as in (6.4).*

To summarize, given a vector $\mathbf{b} \in K[\mathcal{X}]^\ell$ such that $\{b_1, \ldots, b_\ell\}$ is a Gröbner basis (without zero polynomials), the following algorithm determines a set of generators for $\ker \phi_{\mathbf{b}}$.

```
$GroebnerVectorSyzygies(\mathbf{b})$ =
            $S := \emptyset$; $\ell :=$ length($\mathbf{b}$);
            $\mathbf{t} := (\mathrm{lt}(b_i))_{1 \leq i \leq \ell}$;
            for $i$ from 1 to $\ell - 1$ do
              if $b_i \neq 0$ then
                for $j$ from $i + 1$ to $\ell$ do
                  if $b_j \neq 0$ then
                    $(\mathbf{h}, g) := ExtendedReduce(\mathbf{b}, S(b_i, b_j))$;
                # Here, we should have $g = 0$
                $S := S \cup \{\mathbf{v}_{\mathbf{t}}^{ij} - \mathbf{h}\}$
            fi
        od
        fi
    od; return $S$.
```

Example 6.7. We shall determine the output of this algorithm for the vector $\mathbf{b} = (X-Y, Y^3-1)$ corresponding to the Gröbner basis found in Example 3.14 with respect to the lexicographic ordering with $X > Y$. Note that $\mathbf{t} = (X, Y^3)$ and $S(b_1, b_2) = X - Y^4$. Now *ExtendedReduce*$(\mathbf{b}, X - Y^4)$ gives the vector $\mathbf{h} = (1, -Y)$ satisfying $\mathbf{h}\mathbf{b}^\top = X - Y^3$. As the vector of leading terms of $\mathbf{b}$ is $\mathbf{t} = (X, Y^3)$ we have $\mathbf{v}_{\mathbf{t}}^{12} = (Y^3, -X)$, and so the syzygies of $\mathbf{b}$ are spanned by $\mathbf{v}_{\mathbf{t}}^{12} - \mathbf{h} = (Y^3 - 1, Y - X)$.

Syzygies in General

If $\mathbf{a}$ does not correspond to a Gröbner basis, we first need to translate the setting into one in which a Gröbner basis appears. The following lemma takes care of this translation. Consider a $K[\mathcal{X}]$-module morphism $\chi : K[\mathcal{X}]^\ell \to K[\mathcal{X}]$. The result states that the generating set of $\ker \chi$ can be constructed from the generating set of the kernel of another morphism with the same image in $K[\mathcal{X}]$. Recall that, for $\mathbf{a} \in K[\mathcal{X}]^\ell$, we write $\phi_{\mathbf{a}} : K[\mathcal{X}]^\ell \to K[\mathcal{X}]$ for the $K[\mathcal{X}]$-linear map given by $\phi_{\mathbf{a}}(\mathbf{x}) = \mathbf{a}\mathbf{x}^\top$.

Lemma 6.8. *Let* $\mathbf{b} \in K[\mathcal{X}]^\ell$ *and* $\mathbf{a} \in K[\mathcal{X}]^k$. *Suppose F is an $\ell \times k$-matrix over $K[\mathcal{X}]$ satisfying $F\mathbf{a}^\top = \mathbf{b}^\top$ and G is a $k \times \ell$ matrix over $K[\mathcal{X}]$ with $\mathbf{a}^\top = G\mathbf{b}^\top$. If H is a matrix whose rows form a generating set for the $K[\mathcal{X}]$-module* $\ker \phi_{\mathbf{b}}$, *then the kernel of the morphism $\phi_{\mathbf{a}} : K[\mathcal{X}]^k \to K[\mathcal{X}]$ is generated by the rows of the matrices $GF - I$ and HF.*

Proof. As $(GF - I)\mathbf{a}^\top = G\mathbf{b}^\top - \mathbf{a}^\top = 0$ and $(HF)\mathbf{a}^\top = H\mathbf{b}^\top = 0$, clearly all rows of $(GF - I)$ and of HF belong to $\ker \phi_{\mathbf{a}}$.

Conversely, assume $\mathbf{p} = (p_i)_{1 \le i \le k} \in K[\mathcal{X}]^k$ belongs to $\ker \phi_{\mathbf{a}}$. Then $\mathbf{p}G\mathbf{b}^\top = \mathbf{p}\mathbf{a}^\top = 0$, so $\mathbf{p}G \in \ker \phi_{\mathbf{b}}$ and there is $\mathbf{q} \in K[\mathcal{X}]^\ell$ with $\mathbf{p}G = \mathbf{q}H$. Now $\mathbf{p} = \mathbf{p}(I - GF) + \mathbf{p}GF = \mathbf{p}(I - GF) + \mathbf{q}HF$, showing that $\mathbf{p}$ is a $K[\mathcal{X}]$-linear combination of the rows of $GF - I$ and HF.

Returning to the equation $\mathbf{a}\mathbf{x}^\top = 0$, we let $\mathbf{b} \in K[\mathcal{X}]^m$ be a vector corresponding to a Gröbner basis for $\{a_1, \ldots, a_\ell\}$. Then $\phi_{\mathbf{b}}$ and $\phi_{\mathbf{a}}$ have the same image in $K[\mathcal{X}]$, namely the ideal that they generate. Thus there are matrices F and G with $F\mathbf{a}^\top = \mathbf{b}^\top$ and $G\mathbf{b}^\top = \mathbf{a}^\top$. The matrix F can be obtained by keeping track of how the new elements in the Gröbner basis are being formed as $K[\mathcal{X}]$-linear combinations of the old ones in the course of the algorithm *GroebnerBasis*. We shall refer to such a procedure as *ExtendedGroebnerBasis*, so

$$(F, \mathbf{b}) := \mathit{ExtendedGroebnerBasis}(\mathbf{a})$$

yields a vector $\mathbf{b}$ corresponding to a Gröbner basis and a matrix F with $\mathbf{b}^\top = F\mathbf{a}^\top$. The matrix G can be read off from $(G_i, h_i) = \mathit{ExtendedReduce}(\mathbf{b}, a_i)$ for each i. Then G is the matrix whose i-th row is G_i.

We now arrive at the most general version of the Syzygies Algorithm. Given $\mathbf{a} \in K[\mathcal{X}]^k$, the following algorithm determines a set of generators for $\ker \phi_{\mathbf{a}}$.

```
$Syzygies(\mathbf{a})$ =
        $(F, \mathbf{b}) := ExtendedGroebnerBasis(\mathbf{a})$;
        for $i$ from 1 to $k$ do
            $(G_i, g) := ExtendedReduce(\mathbf{b}, a_i)$;
        od;
        $H := GroebnerVectorSyzygies(\mathbf{b})$;
        $S := \emptyset$;
        for $i$ from 1 to $k$ do $S := S \cup \{G_i F - \epsilon_i\}$ od;
        for each $\mathbf{h} \in H$ do $S := S \cup \{\mathbf{h} F\}$ od;
        return $S$.
```

Theorem 6.9. *Upon input of a vector* $\mathbf{a} \in K[\mathcal{X}]^k$, *the algorithm Syzygies computes a set of generators of the* $K[\mathcal{X}]$*-submodule* $\ker \phi_{\mathbf{a}}$ *of* $K[\mathcal{X}]^k$.

Proof. Termination of *Syzygies* is obvious. To see that for $\mathbf{a} \in K[\mathcal{X}]^k$, the algorithm *Syzygies* finds a spanning set for the syzygies of $\mathbf{a}$, observe that $\mathbf{b}$, $\mathbf{a}$, F, G, and H, as computed in the algorithm, all satisfy the hypotheses of Lemma 6.8. So, by the lemma, the rows of the matrices $GF - I$ and HF, which make up the output of *Syzygies*, generate $\ker \phi_{\mathbf{a}}$. Hence the theorem.

Example 6.10. Reconsider $\mathbf{a} = (X^2Y - 1, XY^2 - 1) \in \mathbb{Q}[X, Y]^2$, and recall that, in Example 3.14, we have found $\mathbf{b} = (X - Y, Y^3 - 1)$ such that $\{b_1, b_2\}$ is a Gröbner basis for $\{a_1, a_2\}$. We shall now compute matrices F and G expressing their linear dependencies. We have

$$((Y, -X), X - Y) = \mathit{ExtendedReduce}(\mathbf{a}, S(a_1, a_2)),$$

and,

$$((1, -Y^2), Y^3 - 1) = \mathit{ExtendedReduce}((\mathbf{a}, X - Y), S(a_2, X - Y)).$$

Hence the following $\mathbb{Q}[X, Y]$-linear relations hold:

$$\begin{pmatrix} Y & -X & -1 & 0 \\ 0 & 1 & -Y^2 & -1 \end{pmatrix} \begin{pmatrix} X^2Y-1 \\ XY^2-1 \\ X-Y \\ Y^3-1 \end{pmatrix} = 0.$$

By use of Gauss elimination (elementary row operations), they can be rewritten as $\mathbf{b}^\top = F\mathbf{a}^\top$, where

$$F = \begin{pmatrix} Y & -X \\ -Y^3 & XY^2+1 \end{pmatrix}.$$

In order to find G we just apply $ExtendedReduce(\mathbf{b}, a_i)$ for $i = 1, 2$:

$$\begin{aligned} ExtendedReduce(\mathbf{b}, a_1) &= ((XY+Y^2, 1), 0) \\ ExtendedReduce(\mathbf{b}, a_2) &= ((Y^2, 1), 0) \end{aligned}$$

and so the equation $G\mathbf{b}^\top = \mathbf{a}^\top$ is satisfied for

$$G = \begin{pmatrix} XY+Y^2 & 1 \\ Y^2 & 1 \end{pmatrix}.$$

According to Example 6.7, the matrix H, whose rows are spanning syzygies of $\mathbf{b}$, is $(\, 1-Y^3 \quad X-Y \,)$. Thus, by Lemma 6.8 the syzygies of $\mathbf{a}$ are spanned by the rows of the matrix

$$\begin{pmatrix} GF-I \\ HF \end{pmatrix} = \begin{pmatrix} XY^2-1 & -X^2Y+1 \\ 0 & 0 \\ Y-XY^3 & X^2Y^2-Y \end{pmatrix}.$$

This is obviously a superfluous spanning set. The first row suffices as the third is a scalar multiple of it and the second is trivial. Thus, the extensive computations have led to the simple conclusion that the syzygies of $\mathbf{a}$ are spanned by $(a_2, -a_1)$.

Corollary 6.11. *If K is an effective field, then $K[\mathcal{X}]$ is an effective ring.*

Proof. As stated at the beginning of this section, we only need to verify the existence of an algorithm solving $\mathbf{a}\mathbf{x}^\top = b$ for given $\mathbf{a} \in K[\mathcal{X}]^k$ and $b \in K[\mathcal{X}]$. Let $(F, \mathbf{g}) = ExtendedGroebnerBasis(\mathbf{a})$, so $\mathbf{g}^\top = F\mathbf{a}^\top$. In Corollary 6.6, it has been described how to find a particular solution $\mathbf{e}$ to $\mathbf{g}\mathbf{x}^\top = b$. Then $\mathbf{d} = \mathbf{e}F$ is a particular solution to $\mathbf{a}\mathbf{x}^\top = b$. By Theorem 6.9, the set $S = Syzygies(\mathbf{a})$ is a finite spanning set for the syzygies of $\mathbf{a}$. Therefore (cf. Corollary 6.6) any solution of $\mathbf{a}\mathbf{x}^\top = b$ is of the form

$$\mathbf{d} + \sum_{\mathbf{s} \in S} f_{\mathbf{s}}\mathbf{s} \text{ with } f_{\mathbf{s}} \in K[\mathcal{X}].$$

Corollary 6.12. *If K is an effective field, then each quotient ring of $K[\mathcal{X}]$ is effective.*

Proof. (Sketch) The effectiveness of $K[\mathcal{X}]$ was established in Corollary 6.11. Let us now consider a quotient of $K[\mathcal{X}]$ by an ideal J. By Hilbert's Basis Theorem 1.3, J is finitely generated and so (cf. Theorem 3.16) it is generated by a Gröbner basis $C = \{c_1, \ldots, c_k\}$ with respect to a fixed reduction order $<$.

The elements of the quotient ring $K[\mathcal{X}]/J$ can be represented by polynomials in $K[\mathcal{X}]$ whose monomials are standard monomials, see Proposition 4.4. This also show how equality of two elements can be tested, and how the ring operations can be performed, see the discussion at the beginning of §4.

It remains to verify how we can solve the linear equation in the unknown x_i:

$$\sum_{i=1}^{\ell}(a_i + J)(x_i + J) = b + J \quad \text{with} \quad a_i, b \in K[\mathcal{X}]. \tag{6.5}$$

We may think of $a_1, \ldots, a_\ell, b$ as polynomials in $K[\mathcal{X}]$ which are reduced with respect to C and $<$. Now consider the 'extended' linear equation $\sum_i a_i x_i + \sum_j c_j y_j = b$ in the unknown x_i, y_j. Since $K[\mathcal{X}]$ is effective, we can find $S = \mathit{Syzygies}((\mathbf{a}, \mathbf{c}))$ and $(\mathbf{d}, \mathbf{e}) \in K[\mathcal{X}]^{\ell+k}$ such that any solution in $K[\mathcal{X}]^{\ell+k}$ is of the form

$$(\mathbf{d}, \mathbf{e}) + \sum_{(\mathbf{s},\mathbf{t})\in S} f_{(\mathbf{s},\mathbf{t})}(\mathbf{s}, \mathbf{t}).$$

Now every solution in $K[\mathcal{X}]/J$ to (6.5) is just the projection of a solution as above on the first ℓ coordinates, i.e., of the form

$$\mathbf{d} + \sum_{(\mathbf{s},\mathbf{t})\in S} f_{(\mathbf{s},\mathbf{t})}\mathbf{s}.$$

Exercise 6.13. Suppose the elements $b_1, \ldots, b_\ell$ and $c_1, \ldots, c_k$ generate the respective ideals I and J of K. Prove that a set of generators for the ideal $I \cap J$ can be found effectively by first constructing a generating set G of $\ker \chi$, where $\chi : K^{\ell+k} \to K$ is the morphism sending ϵ_i to b_i if $i \in \{1, \ldots, \ell\}$ and to $c_{i-\ell}$ if $i \in \{\ell + 1, \ldots, \ell + k\}$, and next taking as generating set

$$\left\{ \sum_{i=1}^{\ell} g_i b_i \;\middle|\; g = (g_1, \ldots, g_{\ell+k}) \in G \right\}.$$

Hint: $\chi = \phi_{(\mathbf{b},\mathbf{c})}$.

Notes

Although there is prior work by others (e.g., Hermann, Hironaka, Janet, Zagarias), Gröbner basis theory really got off the ground with Buchberger's thesis [5] (a pleasant introduction to the topic by the same author is [6]). The term Gröbner basis was coined by Buchberger; Gröbner is the name of his thesis adviser. The thesis describes an algorithm for finding a Gröbner basis; the algorithm has later become known as the Buchberger algorithm. The topic has grown into a significant field of mathematics, as can be seen from the proceedings [8].

Recently quite a few introductions to Gröbner basis theory have come to light. The best book we recommend for making a first acquaintance with the topic is [11]. Another good one is [1]. More advanced but very instructive is [24]. Also [4] contains a lot of information. An interesting overview of basic applications is [7].

Implementations of the Buchberger algorithm exist in most general purpose Computer Algebra Packages (*Axiom*, *Magma*, *Maple*, *Mathematica*, *Singular*, *Reduce*, ...). One of the most efficient implementations known to me is *GB* (internet address: http://posso.litp6.fr/GB.html).

Various generalisations of Gröbner bases have not been covered in these notes. First, as mentioned in the introduction, the coefficient ring need not be a field. A treatment of the more general case can be found in [25]. Another generalisation leaves the coefficients in a field, but starts with the noncommutative polynomial ring (cf. [22]) or, even more general, free path algebras (cf. [15]). Complications here are that one cannot expect the analogue of Buchberger's algorithm to exist as the word problem (testing equality when given two representatives) for quotient rings is unsolvable. A common generalization of Gröbner basis theory to both the noncommutative case and arbritrary effective coefficient rings, is presently unknown. There are many special rings for which a Gröbner basis theory is being set up. Here we restrict ourselves to mentioning Lie algebras, for which the notion of Gröbner basis has been worked out and brought into connection with Lyndon-Shirshov bases, see [31, 9].

Section 5 only scratches the surface of what Gröbner bases mean for polynomial system solving. Chapter 2 is devoted to solving polynomial systems with finitely many solutions. There it also becomes clear that other approaches exist. Here we give a few more indications as well. For instance, polynomial systems of equations can also be solved by the use of resultants. See Chapter 9 for the definition of a resultant of two polynomials. Upon input two polynomials f, g in the variables X, Y, it outputs a univariate polynomial in the variable X, called the resultant, with the property that the X-coordinate of each common zero of f and g is a zero of the resultant. Multivariate analogues exist, see [26, 12, 18], but the theory does not seem fully worked out. Yet other algorithms for solving equations involve the involutive method, cf. [27].

The term 'standard monomials' was known long before Gröbner bases. The Grassmannian varieties were examples where reduction of an arbitrary polynomial to a linear combination of standard monomials was well established, see for instance [13, 10].

References

1. **W. W. Adams and P. Loustaunau (1994): *An Introduction to Gröbner Bases*, Graduate Studies in Math. 3, Amer. Math. Soc.**

2. M. F. Atiyah and I. G. Macdonald (1969): *Introduction to Commutative Algebra*, Addison-Wesley, Reading, MA.
3. M. Artin (1991): *Algebra*, Prentice Hall, Englewood Cliffs, NJ.
4. T. Becker and V. Weispfenning (1993): *Gröbner Bases*, Graduate Texts in Mathematics **141**, Springer-Verlag, New York Berlin Heidelberg.
5. B. Buchberger (1965): *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph. D. Thesis, Innsbruck. (An English translation of the journal version of part of this thesis can be found as an appendix in [8].)
6. B. Buchberger (1985): *Gröbner bases: an algorithmic method in polynomial ideal theory*, pp. 184–232 in: Recent Trends in Multidimensional System Theory, N.K. Bose (ed.), Reidel.
7. B. Buchberger (1987): *Applications of Gröbner bases in non-linear computational geometry*, pp. 52–80 in: Trends in Computer Algebra, Lecture Notes in Computer Science **296**, Springer-Verlag, Berlin Heidelberg New York.
8. B. Buchberger and F. Winkler (eds.) (1998): *Gröbner bases and applications*, London Math. Soc. Lecture Note Series **251**, Cambridge University Press, Cambridge.
9. L.A. Bokut and P. Malcolmson (1996): *Gröbner-Shirshov bases for quantum enveloping algebras*, Israel J. Math. **96**, 77–113.
10. A.M. Cohen and R.H. Cushman (1993): *Gröbner bases in standard monomial theory*, pp. 41–60 in: Computational Algebraic Geometry, eds.: F. Eysette and A. Galligo, Progress in Math. **109**, Birkhäuser.
11. D. Cox, J. Little, and D. O'Shea (1992): *Ideals, Varieties, and Algorithms*, Springer-Verlag, Berlin Heidelberg New York.
12. D. Cox and B. Sturmfels (eds.) (1998): *Applications of Computational Algebraic Geometry*, Proceedings of Symposia in Applied Math. **53**, Amer. Math. Soc., Providence, RI.
13. C. DeConcini, D. Eisenbud, and C. Procesi (1980): *Hodge Algebras*, Astérisque **91**.
14. D. Eisenbud (1995): *Commutative Algebra With a View Toward Algebraic Geometry*, Graduate Texts in Math. **150**, Springer-Verlag.
15. Daniel R. Farkas, Feustel, C.D. Green, and L. Edward (1993): *Synergy in the theories of Groebner bases and path algebras*, Can. J. Math. **45**, 727–739.
16. K.O. Geddes, S.R. Czapor, and G. Labahn (1992): *Algorithms for Computer Algebra*, Kluwer, Dordrecht.
17. Green, Edward L. (1993): *An introduction to noncommutative Groebner bases*, pp. 167–190 in: Computational Algebra. Papers from the Mid-Atlantic Algebra Conference, held at George Mason University, Fairfax, VA, USA, May 20-23, 1993. (K.G. Fischer and G. Klaus, eds.) New York: Dekker, Lect. Notes Pure Appl. Math. **151**.
18. I.M. Gelfand, M.M. Kapranov, and A.V. Zelevinsky (1994): *Discriminants, Resultants and Multidimensional Determinants*, Birkhäuser, Basel.
19. R. Hartshorne (1977): *Algebraic Geometry*, Graduate Texts in Mathematics **72**, Springer-Verlag, New York Berlin Heidelberg.
20. M. Kalkbrener (1989): *Solving systems of polynomial equations by using Gröbner bases*, pp. 282–293 in: Proc. EUROCAL '87, Lecture Notes in Computer Science **378**, Springer-Verlag, Berlin Heidelberg New York.
21. P. Lalonde and A. Ram (1995): *Standard Lyndon bases of Lie algebras and enveloping algebras* Trans. Amer. Math. Soc. **347**, 1821–1830.
22. Mora, Teo (1994): *An introduction to commutative and noncommutative Groebner bases*, Theor. Comput. Sci. **134**, 131–173.

23. M. Pohst and H. Zassenhaus (1989): *Algorithmic Algebraic Number Theory*, Encyclopedia of Mathematics and its Applications **30**, Cambridge University Press, Cambridge.
24. L. Robbiano (ed.) (1989): *Computational aspects of commutative algebra* (special issue of Journal of Symbolic Computation), Academic Press, London.
25. W. Trinks (1978): *B. Buchbergers Verfahren, Systeme algebraischer Gleichungen zu lösen*, J. Number Theory **10**, 475–488.
26. B. L. van der Waerden (1931): *Moderne Algebra II*, Springer-Verlag, Berlin Heidelberg New York.
27. A. Yu. Zharkov and Yu. A. Blinkov (1994): Program. Comput. Softw. **20**, 34–36. Translation from Programmirovanie 1994, No. 1, 53–56.

Chapter 2. Symbolic Recipes for Polynomial System Solving

Laureano Gonzalez-Vega, Fabrice Rouillier, and Marie-Françoise Roy

1. Introduction

In many branches of science and engineering where mathematics is used, the resolution of a problem coming from practice is often reduced to the search of a solution for a system of (algebraic or differential) equations modelling the considered problem. From our point of view, to solve a polynomial system of equations is to rewrite it (i.e., to present it in a different form) in such a way that some 'nontrivial' information about its solutions can be derived from this new presentation. The information mentioned above can be related to the existence or non-existence of complex or real solutions, to the number of real or complex solutions, to the approximation of one or several solutions, etc.

The last statement will become clear with the following example: let us consider the following polynomial system:

$$\begin{aligned} f_1 &= x^3 - yx + 1 = 0 \\ f_2 &= x^3 + yx^2 + x - y = 0. \end{aligned}$$

By using some of the techniques to be presented in this chapter, it will be shown that the previous system is equivalent (in the sense that they share the same set of complex solutions) to the following one:

$$\begin{aligned} g_1 &= y^5 - 3y^4 - 4y^3 + 7y^2 - 8y - 2 = 0 \\ g_2 &= x - 2y^4 + 7y^3 + 5y^2 - 17y - 9 = 0. \end{aligned}$$

From this presentation it is very easy to conclude that the initial system has five different complex solutions and that only three of them are real or that

$$(-0.6032, -1.2936)$$

is an approximation for one of the real solutions of the considered system.

The main purpose of this chapter is to show how to use the algorithms and methodology provided by computer algebra to find the solutions of an algebraic system of equations with a finite number of complex solutions. The chapter contains expository parts, which are kept as self-contained and as elementary as possible, recipes, and examples. We try to present the different techniques so that they can be understood by a non-specialist in the subject. The main mathematical prerequisites are elementary algebra (in particular linear algebra).

The chapter is divided into four sections. The first one is devoted to introduce (and to prove) the basic mathematical prerequisites from algebraic geometry needed to understand the rest of the chapter and to show some basic methods in polynomial system solving using Gröbner bases (which are presented in Chapter 1). The second one shows how polynomial system solving can be reduced to a linear algebra problem: the construction of the new system, equivalent to the initial one, is done by working in a concrete finite dimensional vector space and by, mainly, computing efficiently traces of endomorphisms over such vector space. In the next section, we consider the particular case of polynomial systems of equations where the number of equations is equal to the number of unknown. In the last section we consider very briefly numerical approximation techniques where the coordinates of the solutions are presented as the eigenvalues of some of the endomorphisms introduced in the third section. Each of the topics is divided into a theoretical background, a list of recipes, and examples.

2. General Systems of Equations

2.1 Algebraic Preliminaries

Let K be a field of characteristic zero and $\overline{K}$ an algebraically closed field containing it. More concretely, the reader is invited to think of K as being the field $\mathbb{Q}$ of rational numbers and of $\overline{K}$ as being the field $\mathbb{C}$ of complex numbers.

The ring of polynomials in the variables $X_1, \ldots, X_k$ with coefficients in K is denoted by $K[X_1, \ldots, X_k]$ or $K[\mathcal{X}]$, where $\mathcal{X}$ is short for $X_1, \ldots, X_k$. Recall from Chapter 1 that the ring $K[\mathcal{X}]$ can be identified with the ring $K[X_1, \ldots, X_{k-1}][X_k]$. A polynomial in $K[\mathcal{X}]$ is *monic* with respect to X_k if its leading coefficient when considered as a polynomial in X_k with coefficients in $K[X_1, \ldots, X_{k-1}]$ is 1. The *total degree* of a monomial in k variables is the sum of the degrees with respect to each variable and the *total degree* of a polynomial in k variables is the maximum of the total degrees of its monomials.

Let $\mathcal{P}$ be a finite set of polynomials in the variables $X_1, \ldots, X_k$ with coefficients in K and let L be a field containing K as a subfield. The set of *zeros* of $\mathcal{P}$ in L^k is

$$\mathcal{Z}_L(\mathcal{P}) = \{(x_1, \ldots, x_k) \in L^k \mid \forall\, P \in \mathcal{P} \;\; P(x_1, \ldots, x_k) = 0\}.$$

This set is also called the set of *solutions* in L^k of the polynomial system of equations $\mathcal{P} = 0$. Abusing terminology, we also speak of the solutions of a polynomial system $\mathcal{P}$.

To a finite set of polynomials $\mathcal{P}$ is associated the *ideal $I(\mathcal{P})$ generated by $\mathcal{P}$ in $K[\mathcal{X}]$*, i.e., the smallest ideal of $K[\mathcal{X}]$ containing $\mathcal{P}$. In Chapter 1, it was denoted by $(\mathcal{P})$ or $\mathcal{P}K[\mathcal{X}]$. The elements of this ideal are of the form $\sum_{P\in\mathcal{P}} A_P P$ where the A_P are elements of $K[\mathcal{X}]$. A polynomial in $I(\mathcal{P})$ vanishes at any point of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$. Note that when $k=1$, the ideal generated by $\mathcal{P}$ in $K[X_1]$ is *principal* (i.e., generated by a single polynomial), and generated by the gcd of the polynomials in $\mathcal{P}$. We denote the quotient $K[\mathcal{X}]/I(\mathcal{P})$ by A. We can also look at the ideal $\overline{I}(\mathcal{P})$ generated by $\mathcal{P}$ in $\overline{K}[\mathcal{X}]$, i.e., the smallest ideal of $\overline{K}[\mathcal{X}]$ containing $\mathcal{P}$, and define $\overline{A} = \overline{K}[\mathcal{X}]/\overline{I}(\mathcal{P})$. Given $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$ and $Q \in \overline{A}$, the value $Q(x) \in \overline{K}$ is well defined since two polynomials in $\overline{K}[\mathcal{X}]$ having the same image in $\overline{A}$, have the same value at x.

The following well-known result gives an algebraic characterization of systems of polynomials with a finite number of zeros.

Theorem 2.1. *The K-vector space $A = K[\mathcal{X}]/I(\mathcal{P})$ is finite dimensional if and only if $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is a finite set. Moreover, in this case the number of elements of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ does not exceed the dimension of A as a K-vector space.*

The proof of this result relies on a famous theorem in algebra, Hilbert's Nullstellensatz, which we prove for completeness. We first give it in its weak form. The proof we present, due to Michel Coste, is a simplification of Van der Waerden's proof [29].

Theorem 2.2 (Weak Hilbert's Nullstellensatz). *Let $\mathcal{P} = \{P_1, \ldots, P_s\}$ be a finite set of polynomials of $K[\mathcal{X}]$ without zeros in $\overline{K}^k$. There exist polynomials $Q_1, \ldots, Q_s$ of $K[\mathcal{X}]$ such that $1 = P_1Q_1 + \cdots + P_sQ_s$.*

The theorem can be interpreted as follows: it is clear that if an identity $1 = P_1Q_1 + \cdots + P_sQ_s$ holds, the polynomials $P_1, \ldots, P_s$ do not have a common zero. The main point of the theorem is to prove the converse, which is far from obvious.

Proof. The proof is by induction on k. When $k=1$, the ideal generated by $P_1, \ldots, P_s$ in $K[X_1]$ is principal, that is, generated by a single polynomial Q. If Q is not constant, it has a zero in $\overline{K}$ since $\overline{K}$ is algebraically closed, and this zero is common to all P_i.

Suppose now that $k>1$ and that the theorem holds for $k-1$. Since K is infinite, one can suppose that the polynomial P_1 is monic with respect to X_k, after a linear change of variables, using the following lemma.

Lemma 2.3. *Let $P \in K[\mathcal{X}]$ be a polynomial of total degree d. There exists a linear change of coordinates of the form*

$$X_1 = Y_1 + \lambda_1 Y_k, \ldots, \quad X_{k-1} = Y_{k-1} + \lambda_{k-1} Y_k, \quad X_k = Y_k,$$

such that the polynomial $P(Y_1 + \lambda_1 Y_k, \ldots, Y_{k-1} + \lambda_{k-1} Y_k, Y_k)$ is monic with respect to Y_k.

Proof. Consider new indeterminates $\Lambda_1, \ldots, \Lambda_{k-1}$ and take $A(Y_1, \ldots, Y_k) = P(Y_1 + \Lambda_1 Y_k, \ldots, Y_{k-1} + \Lambda_{k-1} Y_k, Y_k)$. The polynomial $A(Y_1, \ldots, Y_k)$ is of degree d in Y_k since the coefficient of Y_k^d in K is a polynomial $B(\Lambda_1, \ldots, \Lambda_{k-1})$ in the variables Λ_i that is not identically zero. It is enough to find elements λ_i of K such that $B(\lambda_1, \ldots, \lambda_{k-1})$ is not zero to prove the lemma. This is always possible in an infinite field, according to the following lemma.

Lemma 2.4. *Let K be an infinite field. If a polynomial $B(Z_1, \ldots, Z_k)$ in $K[Z_1, \ldots, Z_k]$ is not identically zero, there are elements $(z_1, \ldots, z_k)$ in K^k such that $B(z_1, \ldots, z_k)$ is a nonzero element of K.*

Proof. The proof is by induction on k. It is true for a polynomial in one variable since a nonzero polynomial of degree d has at most d roots in a field. Suppose now that it is true for $k-1$ variables, and consider a polynomial $B(Z_1, \ldots, Z_k)$ in k variables which is not identically zero. Thus, if we consider P as a polynomial in Z_k with coefficients in $K[Z_1, \ldots, Z_{k-1}]$, one of its coefficients is not identically zero in $K[Z_1, \ldots, Z_{k-1}]$. Hence, by the induction hypothesis, there exist $(z_1, \ldots, z_{k-1})$ with $B(z_1, \ldots, z_{k-1}, Z_k)$ not identically zero. We are in the case of one variable, which we already considered.

So coming back to the proof of the theorem, suppose that P_1 is monic with respect to X_k. Take a new indeterminate U, and put

$$Q(U, \mathcal{X}) = P_2 + UP_3 + \cdots + U^{s-2} P_s.$$

The resultant (see §1 of Chapter 9) of P_1 and Q with respect to X_k belongs to $K[U, X_1, \ldots, X_{k-1}]$, and is written

$$\mathrm{Res}_{X_k}(P_1, Q) = D_\ell(X_1, \ldots, X_{k-1}) U^\ell + \cdots + D_0(X_1, \ldots, X_{k-1}).$$

This resultant belongs to the ideal generated by P_1 and Q, so there are polynomials Λ and Θ of $K[U, \mathcal{X}]$ such that

$$\mathrm{Res}_{X_k}(P_1, Q) = \Lambda P_1 + \Theta Q.$$

Identifying the coefficients in this equality between polynomials in U, one sees that $D_0, \ldots, D_\ell$ belong to the ideal generated by $P_1, \ldots, P_s$.

Suppose now that $D_0, \ldots, D_\ell$ have a common zero x' in $\overline{K}^{k-1}$. For every $a \in \overline{K}$, we have $\mathrm{Res}_{X_k}(P_1, Q)(a, x') = 0$. Since P_1 is monic with respect to X_k, its leading coefficient in X_k never vanishes, and so the annihilation of the resultant implies that for every $a \in \overline{K}$ the polynomials $P_1(x', X_k)$ and $Q(a, x', X_k)$ have a common root in $\overline{K}$. Since $P_1(x', X_k)$ has a finite number of roots in $\overline{K}$, one of them, say α, is a root of $Q(a, x', X_k)$ for infinitely many $a \in \overline{K}$. Choosing $s-1$ such distinct elements $a_1, \ldots, a_{s-1}$, we get that the polynomial $Q(\alpha, U)$ of degree $\leq s-2$ in U has $s-1$ distinct roots, which is possible only if $Q(x', \alpha, U)$ is identically zero. So one has $P_2(x', \alpha) = \cdots =$

$P_s(x',\alpha) = 0$. Hence (x',α) is a zero of $P_1,\dots,P_s$, which is contrary to the hypothesis.

Thus $D_0,\dots,D_\ell$ have no common zeros in $\overline{K}^{k-1}$. By the induction hypothesis, 1 belongs to the ideal generated by $D_0,\dots,D_\ell$ in $K[X_1,\dots,X_{k-1}]$. As we have seen that $D_0,\dots,D_\ell$ are in the ideal generated by $P_1,\dots,P_s$ in $K[\mathcal{X}]$, we conclude that 1 belongs to this ideal too, which means that there exist polynomials $Q_1,\dots,Q_s$ of $K[\mathcal{X}]$ such that $1 = P_1Q_1 + \cdots + P_sQ_s$.

As usual, Hilbert's Nullstellensatz is derived from the weak form 2.2 using Rabinovitch's trick.

Theorem 2.5 (Hilbert's Nullstellensatz). *Let $\mathcal{P} = \{P_1,\dots,P_s\}$ be a finite set of polynomials with coefficients in K. If a polynomial P with coefficients in K vanishes on the common zeros of $P_1,\dots,P_s$ in $\overline{K}^k$, a power of P belongs to the ideal $I(\mathcal{P})$ in $K[\mathcal{X}]$.*

Proof. The set of polynomials $\{P_1,\dots,P_s,TP-1\}$ has no common zeros in $\overline{K}^{k+1}$ so, according to the weak version of Hilbert's Nullstellensatz (Theorem 2.2), we can find polynomials

$$Q_1(X_1,\dots,X_k,T),\dots,Q_s(X_1,\dots,X_k,T),Q(X_1,\dots,X_k,T)$$

such that $1 = P_1Q_1 + \cdots + P_sQ_s + (TP-1)Q$. Replacing everywhere T by $1/P$ and multiplying by a convenient power of P, we find a power of P in the ideal $I(\{P_1,\dots,P_s\})$.

The set of those polynomials P such that a power of P belongs to the ideal $I(\mathcal{P})$ is called the *radical* of $I(\mathcal{P})$:

$$\sqrt{I(\mathcal{P})} = \{P \in K[\mathcal{X}] \mid \exists\, m \in \mathbb{N} \quad P^m \in I(\mathcal{P})\}.$$

Another usual way of presenting Hilbert's Nullstellensatz is by saying that the radical of $I(\mathcal{P})$ coincides with the set of polynomials in $K[\mathcal{X}]$ vanishing on the common zeros of $P_1,\dots,P_s$ in $\overline{K}^k$.

We now prove Theorem 2.1.

Proof. (Proof of Theorem 2.1) If A is a finite dimensional vector space of dimension N over K, the powers $1, X_1,\dots,X_1^N$ of the variable X_1 are necessarily linearly dependent in A, which gives a polynomial $p_1(X_1)$ in the ideal $I(\mathcal{P})$. This means that the first coordinate of any common zero of $\mathcal{P}$ is a zero of p_1. Doing the same for all the variables, we see that $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is a finite set.

Conversely, if $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is finite, take a polynomial $p_1(X_1) \in \overline{K}[X_1]$ whose zeros are the first coordinates of the elements of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$. According to Hilbert's Nullstellensatz 2.5 a power of p_1 belongs to the ideal $\overline{I}(\mathcal{P})$. Doing the same for all variables we get, for every i, a polynomial of degree d_i in $\overline{K}[X_i]$ in the ideal $\overline{I}(\mathcal{P})$. It means that any monomial of multidegree greater than $(d_1, \ldots, d_k)$ is a linear combination in $\overline{A}$ of monomials of respective degrees in X_i smaller than d_i. Thus, $\overline{A}$ is finite dimensional over $\overline{K}$. We conclude that A is finite dimensional over K using the following lemma.

Lemma 2.6. *Let $\mathcal{P}$ be a finite set of polynomials in $K[\mathcal{X}]$. Then $A = K[\mathcal{X}]/I(\mathcal{P})$ is a finite dimensional vector space of dimension N over K if and only if $\overline{A} = \overline{K}[\mathcal{X}]/\overline{I}(\mathcal{P})$ is a finite dimensional vector space of dimension N over $\overline{K}$.*

Proof. We consider a family $B_1, \ldots, B_m$ of elements of $K[\mathcal{X}]$ and we denote by $b_1, \ldots, b_m$ their images in $K[\mathcal{X}]/I(\mathcal{P})$ and by $\overline{b}_1, \ldots, \overline{b}_m$ their images in $\overline{K}[\mathcal{X}]/\overline{I}(\mathcal{P})$. It is enough to prove that $b_1, \ldots, b_m$ are linearly independent if and only if $\overline{b}_1, \ldots, \overline{b}_m$ are linearly independent. It is clear that if $\overline{b}_1, \ldots, \overline{b}_m$ are linearly independent, $b_1, \ldots, b_m$ are linearly independent. Conversely if $\overline{b}_1, \ldots, \overline{b}_m$ are linearly dependent, it means that there exist $(\lambda_1, \ldots, \lambda_m)$ in $\overline{K}^m \setminus \{0\}$ and, for each $P \in \mathcal{P}$, a polynomial A_P of degree d_P in $\overline{K}[X_1, \ldots, X_k]$ such that

$$\lambda_1 B_1 + \cdots + \lambda_m B_m = \sum A_P P. \qquad (\star)$$

Since the various polynomials $A_P P$ are linear combinations of a finite number of monomials, the identity $(\star)$ can be seen as the fact that a huge linear system of equations with coefficients in K has a solution in $\overline{K}$. We know by linear algebra that this linear system of equations also has solutions in K which means that there are λ_i and A_P with $(\lambda_1, \ldots, \lambda_m)$ in $K^m \setminus \{0\}$, and $A_P \in K[\mathcal{X}]$ with

$$\lambda_1 B_1 + \cdots + \lambda_m B_m = \sum A_P P.$$

This means that $b_1, \ldots, b_m$ are linearly dependent.

Definition 2.7. An element u of A is *separating* if two distinct zeros of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ have different images in $\overline{K}$ by u.

In order to prove the last assertion of the theorem we use the following lemma.

Lemma 2.8. *If $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ has n points, then at least one of the $u_i = X_1 + iX_2 + \cdots + i^{k-1}X_k$ for $0 \le i \le (k-1)\binom{n}{2}$ is separating.*

Proof. Consider a couple $(x, y) = ((x_1, \ldots, x_k), (y_1, \ldots, y_k))$ of distinct points of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ and let $\ell(x, y)$ be the number of i, $0 \le i \le (k-1)\binom{n}{2}$, such that $u_i(x) = u_i(y)$. Since the polynomial $(x_1 - y_1) + (x_2 - y_2)t + \cdots + (x_k - y_k)t^{k-1}$, which is not identically zero, has no more than $k-1$ distinct roots, the number $\ell(x, y)$ is always at most $k - 1$. As the total number of unordered pairs of (distinct) points of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is at most $\binom{n}{2}$, this completes the proof of the lemma.

If u is separating and $\mathcal{Z}_{\overline{K}}(\mathcal{P}) = \{x_1, \ldots, x_n\}$ has n distinct points, then —as we are going to show— $1, u, \ldots, u^{n-1}$ are linearly independent elements of A. Suppose that there exist $\alpha_i \in K$ such that

$$\sum_{i=0}^{n-1} \alpha_i u^i = 0$$

in A; then the polynomial $\alpha_0 + \alpha_1 u^1 + \cdots + \alpha_{n-1} u^{n-1}$ belongs to the ideal $I(\mathcal{P})$ and

$$\sum_{i=0}^{n-1} \alpha_i u^i(x_i) = 0, \quad 1 \leq i \leq n.$$

The univariate polynomial $\sum_{i=0}^{n-1} \alpha_i t^i = 0$ has n distinct roots and is of degree at most $n-1$, hence is the zero polynomial. Thus $1, u, \ldots, u^{n-1}$ are linearly independent elements of A and n is at most the dimension of A as a K-vector space. This concludes the proof of Theorem 2.1.

2.2 First Recipes for Polynomial System Solving

This section is devoted to showing how to use Gröbner bases to solve polynomial systems. Given a reduction ordering (cf. Chapter 1) $<$ on the monoid of all monomials of $K[\mathcal{X}]$, it is possible to define the division of a polynomial $Q \in K[\mathcal{X}]$ by a finite family of polynomials $\mathcal{P}$. This division process is not uniquely defined as there are choices to be made in the process of the computation. Following Chapter 1, a remainder of f with respect to $\mathcal{P}$ (and $<$) will be denoted by *Reduce*$(\mathcal{P}, f)$. Recall that it may happen that a polynomial belongs to the ideal generated by $\mathcal{P}$ while its remainder when divided by $\mathcal{P}$ is not zero.

A *reduced Gröbner basis* of a finite set of polynomials $\mathcal{P}$ for a reduction ordering is defined in Chapter 1. It is a finite set of polynomials $\mathcal{G}$ generating the ideal $I(\mathcal{P})$ with good properties with respect to division and with leading coefficients equal to 1. Namely, the remainder with respect to $\mathcal{G}$ is uniquely determined and a polynomial belongs to the ideal generated by $\mathcal{P}$ if and only if its remainder is zero when divided by $\mathcal{G}$. The quotient $A = K[\mathcal{X}]/I(\mathcal{P})$ is generated as a K-vector space by the monomials *under the staircase*, i.e., the monomials which are not a multiple of the leading monomial of an element in $\mathcal{G}$.

As we have seen in Chapter 1, given a reduction ordering, the Buchberger algorithm is one of the possibilities of computing in a finite number of steps a Gröbner basis for every polynomial system (see also [1, 3, 9, 12, 13, 22]).

The usual reduction orderings (described on the exponents) are:

- *Lexicographic order*: $\alpha = (\alpha_1, \ldots, \alpha_k) >_{\text{lex}} \beta = (\beta_1, \ldots, \beta_k)$ if and only if in the vector difference $\alpha - \beta$ the left-most nonzero entry is positive,

- *Graded lex order*: $\alpha = (\alpha_1, \dots, \alpha_k) >_{\text{grlex}} \beta = (\beta_1, \dots, \beta_k)$ if and only if $\sum \alpha_i > \sum \beta_i$ or $\sum \alpha_i = \sum \beta_i$ and $\alpha >_{\text{lex}} \beta$,
- *Graded reverse lex order*: $\alpha = (\alpha_1, \dots, \alpha_k) >_{\text{grevlex}} \beta = (\beta_1, \dots, \beta_k)$ if and only if $\sum \alpha_i > \sum \beta_i$ or $\sum \alpha_i = \sum \beta_i$ and $\alpha <_{\text{lex}'} \beta$ where lex$'$ is the lexicographical order with $X_k > \dots > X_1$.

The rest of the section is devoted to showing how to obtain information about the zero set of a finite number of polynomials by computing its Gröbner basis.

The first two recipes say that a Gröbner basis computation does not add extraneous solutions to the initial polynomial system and that from the Gröbner basis it is possible to decide whether solutions exist.

Recipe 0: Main Property.

The common zeros of the set of polynomials $\mathcal{P}$ coincide with the common zeros of any Gröbner basis of $\mathcal{P}$ (with respect to any reduction ordering).

Recipe I: Existence of a Zero.

The set of polynomials $\mathcal{P}$ has a common zero in $\overline{K}^k$ if and only if the reduced Gröbner basis of $\mathcal{P}$ with respect to any reduction ordering is not equal to $\{1\}$.

For example, Recipe I shows that the polynomial system of equations

$$\begin{aligned} x^3y - 2xy - 1 &= 0 \\ 3x + y^4x - 2xy &= 0 \\ x^2 - y^2 + 1 &= 0 \end{aligned}$$

has no solution. The next Maple session shows the computation of the Gröbner basis with respect to the reduction ordering $<_{\text{grevlex}}$.

```
List_Pol:=[x**3*y-2*x*y-1,3*x+y**4*x-2*x*y,x**2-y**2+1]:
gbasis(List_Pol,[x,y],tdeg);

                              [1]
```

Recipe II: Normal Form.

Let $\mathcal{G}$ be a reduced Gröbner basis of $\mathcal{P}$ with respect to any reduction ordering $<$. Denote by M the set of monomials under the staircase of $\mathcal{G}$ (recall that these are the monomials which are not divisible by the leading monomial of an element of $\mathcal{G}$). They span a complement to the ideal $I(\mathcal{P})$ in $K[\mathcal{X}]$. The output *Reduce*$(\mathcal{G}, f)$ is unique; it is called the *normal form* of f with respect to ($\mathcal{P}$, $<$) and often abbreviated to $NF(f)$. The map $f \mapsto NF(f)$ is a function from $K[\mathcal{X}]$ into the set of linear combinations of elements of M. The main property of this function is the following (cf. Chapter 1):

A polynomial h belongs to the ideal generated by $\mathcal{P}$ if and only if $NF(h) = 0$.
The normal form $NF(h)$ of h is the unique linear combination of elements of M such that $h - NF(h)$ belongs to the ideal generated by $\mathcal{P}$.

The next example describes how to compute the monomials under the staircase and the normal form.

Example 2.9. Consider the polynomial system of equations:

$$P_1 := x^2y - 2x^2 + y^2 + xy = 0 \qquad P_2 := 2x^2 - y^2 + xy = 0.$$

The Gröbner basis of P_1 and P_2 with respect to the reduction ordering $<_{\text{grevlex}}$ is:

$$\mathcal{G} := \{2x^2 - y^2 + xy, y^4 + 6y^3 + 16xy, xy^2 - 4xy - y^3\}.$$

The leading terms of the polynomials in $\mathcal{G}$ are computed below.

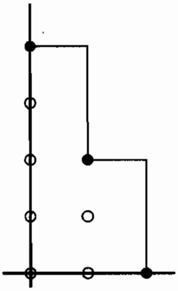
```
f1:=x**2*y-2*x**2+y**2+x*y: f2:=2*x**2-y**2+x*y:
GB:=gbasis([f1,f2],[x,y],tdeg);
```

$$[2x^2 - y^2 + xy, -4xy - y^3 + xy^2, y^4 + 6y^3 + 16xy]$$

```
for j from 1 to 3 do print(leadmon(GB[j],[x,y],tdeg)) od;
```

$$[2, x^2],\ [1, xy^2],\ [1, y^4]$$

The set of monomials under the staircase



is a finite set of monomials consisting of those monomials which are not multiples of y^4, xy, or x^2:

$$\mathcal{A} = \{1, y, y^2, y^3, x, xy\} = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}.$$

If we ask for the normal form of a polynomial in x and y (using the function **normalf** of Maple) we find a linear combination of elements in $\mathcal{A}$. For example the normal form of x^3 is computed as follows:

```
normalf(x**3,GB,[x,y],tdeg);
```

$$(1/2)y^3 + 3xy.$$

`Recipe III`: **Existence of a Finite Number of Zeros.**

The set of polynomials $\mathcal{P}$ has a finite number of zeros if and only if the reduced Gröbner basis $\mathcal{G}$ of $\mathcal{P}$ with respect to any reduction ordering has the following property:

> For every unknown X_i there exists a polynomial in $\mathcal{G}$ such that its leading term with respect to the considered reduction ordering (the biggest of its monomials) is equal to X_i^s for some $s > 0$.

The explanation of this recipe relies on the fact that A is a finite dimensional vector space if and only if the set of monomials under the staircase is finite.

The number of solutions of the polynomial system of equations

$$\begin{aligned} z^3y - 2xy + z &= 0 \\ 3xz + y^4x - 2z^2 &= 0 \\ z^3y - 2xy + z - 3xz - y^4x + 2z^2 &= 0 \end{aligned}$$

is infinite. Its Gröbner basis with respect to $<_{\text{grevlex}}$ is:

$$[z^3y - 2\,xy + z, 3\,xz + y^4x - 2\,z^2, 6\,x^2z - 4\,xz^2 + y^3xz - 3\,z^4x + 2\,z^5,$$
$$-2\,x^2y^3 + xy^2z - 6\,z^3x^2 + 4\,z^4x + 3\,z^6x - 2\,z^7],$$

so that, by `Recipe III`, the number of solutions is infinite since there does not exist a polynomial in the Gröbner basis such that its initial term with respect to $<_{\text{grevlex}}$ is a power of x.

The next recipe shows how to transform the initial polynomial system of equations (with a finite number of solutions) into a triangular system.

`Recipe IV`: **Reduction to Triangular Form.**

If a finite set of polynomials has a finite number of common zeros then its reduced Gröbner basis with respect to the reduction ordering $<_{\text{lex}}$ has the following structure:

$$\begin{array}{ll} X_k^{s_k} + w_k(X_k) & \deg_{X_k}(w_k) < s_k \\ X_{k-1}^{s_{k-1}} + w_{k-1}(X_{k-1}, X_k) & \deg_{X_{k-1}}(w_{k-1}) < s_{k-1} \\ \vdots & \vdots \\ X_1^{s_1} + w_1(X_1, X_2, \ldots, X_{k-1}, X_k) & \deg_{X_1}(w_1) < s_1 \\ Q_1(X_1, \ldots, X_k), \ldots, Q_t(X_1, \ldots, X_k). & \end{array}$$

More information about how Gröbner bases provide triangular systems can be found in [15, 21, 25].

In the case of the polynomial system of equations

$$\begin{aligned} x^2+y+z-1&=0\\ x+y^2+z-1&=0\\ x+y+z^2-1&=0 \end{aligned}$$

we get:

$$[x+y+z^2-1, y^2+z-y-z^2, -z^2+2\,yz^2+z^4, -z^2-4\,z^4+z^6+4\,z^3],$$

which provides a single polynomial in one variable, two polynomials in two variables and one polynomial in three variables:

$$\begin{cases} x+y+z^2-1=0 \\ \quad\begin{cases} -y-z^2+y^2+z=0 \\ 2z^2y-z^2+z^4=0 \end{cases} \\ \qquad -z^2-4z^4+z^6+4z^3=0. \end{cases}$$

The next recipe shows that under suitable assumptions the Gröbner basis with respect to $<_{\text{lex}}$ has a very simple structure. The initial system is reduced to a univariate equation.

Recipe V: Shape Lemma.

If a polynomial system of equations has a finite number of solutions, and moreover the variable X_k is separating and all solutions are regular (see Definitions 2.7 and 3.2), its Gröbner basis with respect to the reduction ordering $<_{\text{lex}}$ has the following structure:

$$X_1-g_1(X_k), X_2-g_2(X_k), \ldots, X_{k-1}-g_{k-1}(X_k), g_k(X_k),$$

where every $g_i(X_k)$ is a univariate polynomial.

We say that we are in the Shape Lemma case if this structure holds (see [5] for a more detailed discussion about this condition).

The Shape Lemma provides a very simple structure: the problem has been reduced to the resolution of a single equation in one unknown. For example, for the polynomial system of equations

$$\begin{aligned} x^3+yx^2-y^2 &=0\\ 2xy^2+x^2-y+1 &=0 \end{aligned}$$

we find the following system:

$$[2y^4+y^2+y-1+x+4y^5, 4y^2-3y+1-4y^7+4y^5-3y^3+8y^8-2y^4].$$

Nevertheless it is not always the case that the Gröbner basis with respect to $<_{\text{lex}}$ has such a simple shape. For example, if we consider the system

$$\begin{aligned}1 - x - xy^2 - xz^2 &= 0\\ 1 - y - yx^2 - yz^2 &= 0\\ 1 - z - zx^2 - zy^2 &= 0,\end{aligned}$$

the Gröbner basis with respect to $<_{\text{lex}}$ does not have the Shape Lemma structure:

$$\begin{aligned}&6322x + 6322y + 7288z^{13} - 9070z - 1599 - 33328z^{12} - 2877z^4 + 120322z^9\\ &\quad -146680z^8 + 22343z^2 - 21246z^3 - 13493z^5 + 80725z^7 - 73097z^6\\ &\quad -111028z^{10} + 61924z^{11},\end{aligned}$$

$$\begin{aligned}&12644y^2 + 25288yz^6 + 37932yz^4 + 12644z^3y + 12644yz^2 + 12644zy - 12644y\\ &\quad -72904z^{13} + 14276z + 104160z^{12} + 13497 - 95537z^4 - 476038z^9 + 197484z^8\\ &\quad -33795z^2 + 50108z^3 - 85671z^5 - 345531z^7 - 55049z^6 + 260236z^{10}\\ &\quad -296668z^{11},\end{aligned}$$

$$\begin{aligned}&8yz^7 + 20z^5y + 4yz^4 + 16z^3y + 8yz^2 + 4y + 8z^{13} - 6z - 16z^{12} - 3z^4 + 62z^9\\ &\quad -48z^8 + 5z^2 - 20z^3 + z^5 + 45z^7 - 19z^6 - 44z^{10} + 36z^{11} - 1,\end{aligned}$$

$$\begin{aligned}&-z + 28z^{12} - 3z^4 - 8z^{13} - 2z^9 + 31z^8 - z^2 - z^3 + 12z^5 + 8z^{14} + 20z^7 + 12z^6\\ &\quad +42z^{10} - 16z^{11} - 1.\end{aligned}$$

In the Shape Lemma case, a lexicographic Gröbner basis gives a univariate polynomial $f \in K[X_1]$ such that A is isomorphic to $K[X_1]/I(\{f\})$. This is the simplest form we can hope to find for the quotient $A = K[\mathcal{X}]/I(\mathcal{P})$; nevertheless, there are cases where $K[\mathcal{X}]/I(\mathcal{P})$ is not isomorphic to a finite algebra of this form. For example, it is easy to show that $K[X_1, X_2]/I(\{X_1^2, X_1X_2, X_2^2\})$ is not of this form.

The computation of a Gröbner basis using the Buchberger algorithm depends strongly on the chosen reduction ordering. In particular, in many examples, the computation of the Gröbner basis with respect to $<_{\text{lex}}$ requires an enormous computing time or the size of the output is too big, while computing a Gröbner basis for another reduction ordering is possible.

When there are a finite number of zeros, the lexicographic Gröbner basis can be computed efficiently by a change of ordering (**FGLM** Algorithm — see [14]). Given a Gröbner basis for another reduction ordering, standard linear algebra algorithms can be easily applied in the K-vector space A. The lexicographic Gröbner basis is obtained by detecting linear combinations of monomials in A. For example, the first polynomial $f_k(X_k)$ can be viewed as the minimal polynomial of X_k. More generally, we put successively in an $N \times N$ matrix (N is the dimension of A), the reduced expressions of possible monomials (considering them with respect to the lexicographic ordering and starting from 1), removing the ones that can be expressed as a linear combination (each combination gives one polynomial of the lexicographic basis — remark that the first found combination is the minimal polynomial of X_k) of the preceding ones and stopping when the matrix has full rank.

3. Linear Algebra, Traces, and Polynomial Systems

In this section we consider a finite set of polynomials $\mathcal{P}$ such that $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is finite, of size, say n, so that with the notation of the last section, A and $\overline{A}$ are finite dimensional vector spaces over K and $\overline{K}$ of the same dimension, say N. We are going to indicate the relations between the solutions of the system of equations and the eigenvalues of specific linear endomorphisms of A and $\overline{A}$.

3.1 Eigenvalues and Polynomial Systems

We are going to explain how the quotient ring $\overline{A}$ splits in a finite number of local factors, one for each solution of the polynomial system. These local factors are used to define the multiplicities of the solutions of the polynomial system. In the most usual case these local factors will be nothing but the field $\overline{K}$ itself, and the splitting will consist of sending an element of $\overline{A}$ to its values at the various points of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$.

We need a new definition. A *local ring* B is a ring such that for every $a \in B$, either a is invertible or $1 + a$ is invertible. A field is always a local ring. An equivalent definition of local ring is a ring with a unique maximal (proper) ideal —the set of non-invertible elements then forms the maximal ideal.

Given a multiplicative subset S of a ring A (i.e., a subset of A closed under multiplication), we define an equivalence relation on couples (a, s) with $a \in A$ and $s \in S$ by $(a, s) \sim (a', s')$ if and only if there exists $t \in S$ such that $t(as' - a's) = 0$. The class of (a, s) is denoted by a/s. The *ring of fractions* $S^{-1}A$ is the set of classes a/s equipped with the following addition and multiplication

$$(a/s) + (a'/s') = (as' + a's)/(ss'), \quad (a/s)(a'/s') = (aa')/(ss').$$

The *localization of* $\overline{A}$ *at* $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$, denoted by $\overline{A}_x$, is the ring of fractions associated to the multiplicative subset S_x consisting of elements of $\overline{A}$ not vanishing at x. The ring $\overline{A}_x$ is local: an element P/Q of $\overline{A}_x$ is invertible if and only if $P(x) \neq 0$, and it is clear that either P/Q is invertible or $1 + (P/Q) = (Q + P)/Q$ is invertible.

We are going to prove the following result.

Theorem 3.1.

$$\overline{A} \cong \prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \overline{A}_x.$$

Definition 3.2. We denote by $\mu(x)$ the dimension of $\overline{A}_x$ as a $\overline{K}$-vector space. We call $\mu(x)$ the *multiplicity* of the zero $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$. If $\mu(x)$ is 1, then x is said to be a *regular* zero of $\mathcal{P}$.

Every element $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$ is regular if and only if $\overline{A}$ is a product of a finite number of copies of $\overline{K}$.

We are also going to prove the next result, less well known but extremely useful.

Theorem 3.3 (Stickelberger's Theorem). *Let $f \in A$ and let L_f be the linear endomorphism of multiplication by f (so that $L_f(g) = fg$ for $g \in A$). Then $L_f(\overline{A}_x) \subset \overline{A}_x$. The restriction of L_f to $\overline{A}_x$ has only one eigenvalue $f(x)$; its multiplicity is $\mu(x)$.*

The proof of Theorem 3.1 is based on the following result.

Proposition 3.4. *If $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ is finite, then, for every $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$, there exist elements e_x of $\overline{A}$ with*

- ○ $\displaystyle\sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} e_x = 1,$
- ○ $e_x^2 = e_x,$
- ○ $e_x(x) = 1,$
- ○ $e_x e_y = 0$ *for* $y \neq x$ *with* $y, x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$.

Proof. We first prove that there exist elements s_x of $\overline{A}$ ($x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$) with $s_x(x) = 1, s_x(y) = 0$ for every $y \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$, $y \neq x$. Without loss of generality we can suppose that the variable X_1 is separating. The classical Lagrange interpolation gives polynomials in X_1 with the required properties.

Since $s_x s_y$ vanishes on every common zero of $\mathcal{P}$, Hilbert's Nullstellensatz 2.5 implies that there exist powers of s_x, denoted by t_x, such that $t_x t_y = 0$ in $\overline{A}$ for $y \neq x$, and $t_x(x) = 1$. The family of polynomials $\mathcal{P} \cup \{t_x \mid x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})\}$ has no common zeros so, according to Hilbert's Nullstellensatz, there exist polynomials r_x such that $\sum t_x r_x = 1$ in $\overline{A}$. Take $e_x = t_x r_x$. It is easy to verify the claimed properties.

The element e_x is called the *idempotent* attached to x. Since e_x is idempotent, $e_x\overline{A}$ equipped with the restriction of the addition and multiplication of A is a ring.

We prove now that the ring $e_x\overline{A}$ coincides with the localization of $\overline{A}$ at x. The isomorphism is as follows: the element $e_x P$ of $e_x\overline{A}$ is sent to $e_x(P/1)$, the element (P/Q) of $\overline{A}_x$ is sent to $P(1/Q(x)(1 - v + \ldots + (-v)^{N-1}))e_x$ where v is defined by $Q = Q(x)(1 + v)$. To see that this is an isomorphism, note that ve_x is zero everywhere on $\mathcal{Z}_{\overline{K}}(\mathcal{P})$, so that $(ve_x)^N = v^N e_x = 0$. Thus, $(1 + v)e_x$ is invertible in $e_x\overline{A}$ and its inverse is $(1 - v + \ldots + (-v)^{N-1})e_x$, so that $Q(1/Q(x)(1-v+\ldots+(-v)^{N-1}))e_x = e_x$, and $(P/Q) = Pe_x(1/Q(x)(1-v \ldots + (-v)^N))$.

Since$\sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} e_x = 1$, $\overline{A} \cong \prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \overline{A}_x$. The canonical surjection of $\overline{A}$ onto $\overline{A}_x$ coincides with the multiplication by e_x.

This ends the proof of Theorem 3.1. More precisely, we have shown

Theorem 3.5. *For every $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$ there exists an idempotent e_x such that $e_x\overline{A} = \overline{A}_x$ and*

$$\overline{A} \cong \prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \overline{A}_x.$$

We now come to the conclusion of the proof of Stickelberger's Theorem 3.3.

Proof (of Theorem 3.3). Since $\overline{A}_x$ is the image of $\overline{A}$ under multiplication by e_x, it is clear that $L_f(\overline{A}_x) \subset \overline{A}_x$. As $e_x(f - f(x))$ vanishes on the common zeros of $\mathcal{P}$, Hilbert's Nullstellensatz implies that there exists $m \in \mathbb{N}$ such that $(e_x(f - f(x)))^m = 0$, which means that $L_{f-f(x)}$ is nilpotent. This proves Stickelberger's Theorem.

If the multiplicity of x is 1, then, as a consequence of the preceding result, $\overline{A}_x = \overline{K}$ and the canonical surjection $\overline{A} \to \overline{A}_x$ coincides with evaluation at x.

Corollary 3.6. *For $f \in A$, the multiplication endomorphism L_f has the following properties:*

- *The trace of L_f is*

$$\sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \mu(x) f(x).$$

- *The determinant of L_f is*

$$\prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} f(x)^{\mu(x)}.$$

- *The characteristic polynomial of L_f is*

$$\chi_f(T) = \prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} (T - f(x))^{\mu(x)}.$$

3.2 Counting Solutions and Removing Multiplicities

Since A is a finite dimensional vector space, any endomorphism of A can be represented by means of a matrix with respect to a fixed basis. For every $h \in A$, we define the *h-trace bilinear form*, notation TrB_h, (or simply trace, notation TrB, if $h = 1$) as the bilinear map:

$$\begin{array}{rccc} \mathrm{TrB}_h\colon & A \times A & \longrightarrow & K \\ & (f, g) & \longmapsto & \mathrm{Trace}(L_{fgh}), \end{array}$$

where Trace denotes the usual trace of a linear endomorphism. The corresponding quadratic form associated to TrB_h, called *Hermite quadratic form* will be denoted by:

$$\begin{array}{rccc} Q_h\colon & A & \longrightarrow & K \\ & f & \longmapsto & \mathrm{Trace}(L_{f^2h}). \end{array}$$

The main properties of these quadratic forms are summarized in the next two theorems. We shall write A_{red} to denote the algebra $K[\mathcal{X}]/\sqrt{I(\mathcal{P})}$.

The next theorem gives the connection between TrB, the radical of $I(\mathcal{P})$ and the kernel of the quadratic form Q_1:

$$\ker(Q_1) = \{f \in A \mid \forall\, g \ \ \mathrm{TrB}(f,g) = 0\}.$$

Theorem 3.7.

$$f \in \sqrt{I(\mathcal{P})} \qquad \Longleftrightarrow \qquad f \in \ker(Q_1).$$

Proof. Let f be an element of $\sqrt{I(\mathcal{P})}$. Then f vanishes on every element of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$. So, applying Corollary 3.6, we obtain the following equality for every $g \in K[\mathcal{X}]$:

$$\mathrm{TrB}(f,g) = \sum_{i=1}^{n} \mu(x_i) f(x_i) g(x_i) = 0.$$

Conversely, if f is an element such that, for any g in A, $\mathrm{TrB}(f,g) = 0$ then Corollary 3.6 gives:

$$\mathrm{TrB}(f,g) = \sum_{i=1}^{n} \mu(x_i) f(x_i) g(x_i) = 0 \qquad \forall\, g \in A, \tag{$\star$}$$

where $x_1, \ldots, x_n$ are the elements of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$. Let u be a separating element. Equality ($\star$) applied with $g = 1, \ldots, u^{n-1}$ gives:

$$\begin{pmatrix} 1 & \ldots & 1 \\ \vdots & & \vdots \\ u(x_1)^{n-1} & \ldots & u(x_n)^{n-1} \end{pmatrix} \cdot \begin{pmatrix} \mu(x_1) f(x_1) \\ \vdots \\ \mu(x_n) f(x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

so that $f(x_1) = \ldots = f(x_n) = 0$, since u is separating and the matrix at the lefthand-side is a Vandermonde matrix, whence invertible. Using Hilbert's Nullstellensatz 2.2, we obtain $f \in \sqrt{I(\mathcal{P})}$ as wanted.

The rank of Q_h gives interesting information.

Theorem 3.8. *For $h \in A$, the quadratic form Q_h satisfies:*

$$\mathrm{rank}(Q_h) = \#\{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P}) \mid h(x) \neq 0\}.$$

Proof. Consider a separating element u. The elements $1, u, \ldots, u^{n-1}$ are linearly independent and can be completed to a basis

$$\omega_1 = 1, \omega_2 = u, \ldots, \omega_n = u^{n-1}, \omega_{n+1}, \ldots, \omega_N$$

of the K-vector space A. Corollary 3.6 provides the following expression for the quadratic form Q_h:

$$g=\sum_{j=1}^{N} g_j\omega_j \in A \quad\Longrightarrow\quad Q_h(g)=\sum_{i=1}^{n}\mu(x_i)h(x_i)\left(\sum_{j=1}^{N} g_j\omega_j(x_i)\right)^2,$$

where the x_i's are the elements in $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ and the $\mu(x_i)$'s are the corresponding multiplicities. Consequently, Q_h is the map

$$g\mapsto (g_1,\dots,g_N)^\top\cdot\Gamma^\top\cdot\Delta(\mu(x_1)h(x_1),\dots,\mu(x_n)h(x_n))\cdot\Gamma\cdot(g_1,\dots,g_N)$$

where

$$\Gamma=\begin{pmatrix}1 & \dots & u(x_1)^{n-1} & \omega_{n+1}(x_1) & \dots & \omega_N(x_1)\\ \vdots & & & & & \vdots\\ 1 & \dots & u(x_n)^{n-1} & \omega_{n+1}(x_n) & \dots & \omega_N(x_n)\end{pmatrix}$$

and Δ denotes a diagonal matrix with indicated diagonal entries. Therefore it suffices to prove that the rank of Γ is equal to n. But u is separating and the principal minor of the matrix Γ is a Vandermonde determinant.

If $\mathcal{A}=\{\omega_1,\dots,\omega_N\}$ is a basis of A, then the matrix of TrB_h with respect to $\mathcal{A}$ is given by:

$$\begin{pmatrix}\mathrm{Trace}(h\omega_1\omega_1) & \dots & \mathrm{Trace}(h\omega_1\omega_N)\\ \vdots & & \vdots\\ \mathrm{Trace}(h\omega_N\omega_1) & \dots & \mathrm{Trace}(h\omega_N\omega_N)\end{pmatrix},$$

where $\mathrm{Trace}(h\omega_i\omega_j)$ represents the trace of the endomorphism $L_{h\omega_i\omega_j}$. In what follows the matrix of L_h with respect to the considered basis $\mathcal{A}$ will be denoted by $\mathcal{L}_h$. If $h=1$ then the matrix TrB_1 is simply called *trace matrix* and denoted by TrM.

Recipe VI: Counting Solutions and Removing Multiplicities.

- ∘ $\mathrm{rank}(Q_1)=\#(\mathcal{Z}_{\overline{K}}(\mathcal{P}))=n$,
- ∘ $\mathrm{rank}(Q_h)=\#\{x\in\mathcal{Z}_{\overline{K}}(\mathcal{P})\mid h(x)\neq 0\}$,
- ∘ If ker (TrM) is generated by $\{g_1,\dots,g_t\}$ as a linear subspace of A then $\mathcal{P}\cup\{g_1,\dots,g_t\}$ is a system of generators for $\sqrt{I(\mathcal{P})}$, i.e., a polynomial system with the same solutions as the initial one but such that all the solutions are regular according to Theorem 3.7.

The last item of this recipe allows us to apply numerical methods, such as Homotopy Methods or the Newton Method, more safely for approximating the solutions.

Example 3.9 (Continuation of Example 2.9). Determining the Trace Matrix TrM requires the computation of the trace of the matrices of multiplication of any two basis elements. For example

$$\mathcal{L}_{\omega_3\omega_6} = \begin{pmatrix} 0 & * & * & * & * & * \\ * & 0 & * & * & * & * \\ * & * & 0 & * & * & * \\ * & * & * & 176 & * & * \\ * & * & * & * & 0 & * \\ * & * & * & * & * & -64 \end{pmatrix}$$

(we only give the values of the diagonal elements of the matrix). Its trace is 112.

These computations produce:

$$\mathrm{TrM} = \begin{pmatrix} 6 & -2 & 20 & -56 & -4 & 4 \\ -2 & 0 & -56 & 272 & 4 & -40 \\ 20 & -56 & 272 & -992 & -40 & 112 \\ -56 & 272 & -992 & 4160 & 112 & -544 \\ -4 & 4 & -40 & 112 & 8 & -8 \\ 4 & -40 & 112 & -544 & -8 & 80 \end{pmatrix}$$

(112 appears at the entry in the sixth row and the third column). We conclude that rank(TrM) $= 3$ and $\sqrt{I(\mathcal{P})}$ is generated by the polynomials

$$\{P_1, P_2, 2y + xy, 2y + y^2 + 4x\}.$$

More information about how to transform the resolution of a polynomial system of equations to a linear algebra problem can be found in [3, 26, 27, 28, 30, 31].

3.3 Rational Univariate Representation

In this section we describe a method, based on trace computations and known as the rational univariate representation ([2, 28]), for solving a polynomial system of equations. We are going to describe the coordinates of the solutions of a polynomial system with a finite number of solutions as rational functions of the roots of a univariate polynomial. Compared to the Shape Lemma method described above, this method is completely general. Even in the Shape Lemma case, it gives a more compact description of the solutions.

3.3.1 Definition and Properties. For any element $u \in A$, let $\chi_u(T)$ be the characteristic polynomial of the linear transformation L_u. Then, according to Corollary 3.6,

$$\chi_u(T) = \prod_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} (T - u(x))^{\mu(x)}.$$

The polynomial $\chi_u(T)$ can be computed with the following method. The i-th *Newton sum* s_i (cf. Chapter 5) associated to the polynomial $\chi_u(T)$ is by definition

$$s_i = \sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \mu(x) u(x)^i.$$

According to Stickelberger's Theorem 3.3, $s_i = \mathrm{Trace}(u^i)$. If

$$\chi_u(T) = \sum_{i=0}^{N} b_i T^{N-i},$$

then

$$\frac{\chi_u'(T)}{\chi_u(T)} = \sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \frac{\mu(x)}{T - u(x)} = \sum_{j \geq 0} \frac{\mathrm{Trace}(u^j)}{T^{j+1}}$$

and thus

$$\chi_u'(T) = \sum_{k=0}^{N-1} \sum_{j=0}^{N-k-1} \mathrm{Trace}(u^j) b_k T^{N-k-j-1}.$$

Identifying the coefficients of T^{N-i-1} on both sides, we get *Newton's formula* (cf. Chapter 5):

$$(N-i)b_i = \sum_{j=0}^{i} \mathrm{Trace}(u^j) b_{i-j}, \tag{3.1}$$

so that $\chi_u(T)$ can be computed from $\mathrm{Trace}(u^j)$, for $j = 0, \ldots, N$.

For any $v \in A$, we define:

$$g_u(v,T) = \sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \mu(x) v(x) \prod_{\nu \in u(\mathcal{Z}_{\overline{K}}(\mathcal{P})), \nu \neq u(x)} (T - \nu).$$

If u is separating, the multiplicity $\mu(x)$ of x coincides with the multiplicity μ of $u(x)$ as a root of $\chi_u(T)$ and we can express the values of v at the points of the set $\mathcal{Z}_{\overline{K}}(\mathcal{P})$ as rational functions of the roots of $\chi_u(T)$ $(z \in \mathcal{Z}_{\overline{K}}(\mathcal{P}))$, since:

$$\frac{g_u(v,u(z))}{g_u(1,u(z))} = \frac{\displaystyle\sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \mu(x) v(x) \prod_{\nu \in u(\mathcal{Z}_{\overline{K}}(\mathcal{P})), \nu \neq u(x)} (u(z) - \nu)}{\displaystyle\sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \mu(x) \prod_{\nu \in u(\mathcal{Z}_{\overline{K}}(\mathcal{P})), \nu \neq u(x)} (u(z) - \nu)} = v(z).$$

We indicate now how to express $g_u(v,T)$, following [2]. Given a monic polynomial P, we denote by $\overline{P}$ the squarefree part of P, i.e., the monic polynomial $\overline{P} = P/\gcd(P,P')$. Note that, since

$$\overline{\chi}_u(T) = \prod_{\nu \in u(\mathcal{Z}_{\overline{K}}(\mathcal{P}))} (T - \nu)$$

is the squarefree part of $\chi_u(T)$,

$$\frac{g_u(v,T)}{\overline{\chi}_u(T)} = \sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \frac{\mu(x)v(x)}{T - u(x)} = \sum_{i \geq 0} \frac{\displaystyle\sum_x \mu(x)v(x)u(x)^i}{T^{i+1}} = \sum_{j \geq 0} \frac{\mathrm{Trace}(vu^j)}{T^{j+1}}.$$

If

$$\overline{\chi}_u(T) = \sum_{i=0}^{n} a_i T^{n-i},$$

multiplying both sides by $\overline{\chi}_u(T)$ and using that $g_u(v, T)$ is a polynomial in $\overline{K}[T]$, we have:

$$g_u(v,T) = \sum_{k=0}^{n-1} \sum_{j=0}^{n-k-1} \mathrm{Trace}(vu^j) a_i T^{n-k-j-1}.$$

This equality proves that $g_u(v, T) \in K[T]$.

Given a univariate polynomial $p = \sum_{i=0}^{n} c_i T^{n-i}$ we denote, for $j = 0, \dots, n$, the j-th Horner polynomial associated to p by

$$H_j(p) = \sum_{i=0}^{j} c_i T^{j-i}.$$

The expression of $g_u(v, T)$ becomes:

$$g_u(v,T) = \sum_{j=0}^{n-1} \mathrm{Trace}(vu^j) H_{n-j-1}(\overline{\chi}_u(T)).$$

This last equality provides a method for computing $g_u(v, T)$ from $\chi_u(T)$ and $\mathrm{Trace}(vu^j)$ for $j = 0, \dots, n$.

Let $\mathcal{P}$ be a set of polynomials with a finite number of common zeros. Let $u \in A$ be a separating element. The set of polynomials

$$\{\chi_u(T), g_u(1,T), g_u(X_1,T), \dots, g_u(X_k,T)\}$$

defines the *rational univariate representation* of $\mathcal{P}$ associated to u. The following proposition summarizes its most important properties (see [28]).

Proposition 3.10. *The rational univariate representation associated to a separating element u satisfies the following properties:*

1. *The polynomials $\chi_u(T), g_u(1,T), g_u(X_1,T), \dots, g_u(X_k,T)$ are elements of $K[T]$.*
2. *The degree of the squarefree part of $\chi_u(T)$ is equal to the number of elements in $\mathcal{Z}_{\overline{K}}(\mathcal{P})$.*
3. *If x is a root of the system, $u(x)$ is a root of $\chi_u(T)$ with the same multiplicity. Conversely, if t is a root of $\chi_u(T)$,*

$$\left(\frac{g_u(X_1,t)}{g_u(1,t)}, \dots\dots, \frac{g_u(X_k,t)}{g_u(1,t)}\right)$$

is a root of the system with the same multiplicity.

4. *In the Shape Lemma case, $u = X_k$ is separating and $g_u(1, X_k)$ is an invertible element of A; the lexicographic Gröbner basis can be derived from the rational univariate representation associated with X_k:*

$$\begin{cases} X_1 - (g_u(1,X_k)^{-1} g_u(X_1,X_k) \bmod \chi_u(X_k)) \\ \qquad\qquad\qquad\qquad \vdots \\ X_{k-1} - (g_u(1,X_k)^{-1} g_u(X_{k-1},X_k) \bmod \chi_u(X_k)) \\ \chi_u(X_k). \end{cases}$$

5. *The rational univariate representation (associated to u) of the radical of $I(\mathcal{P})$ is, with $\overline{g}_u(v,T)$ denoting the polynomial $g_u(v,T)$ mod $\overline{\chi}_u(T)$,*

$$\{\overline{\chi}_u(T), \overline{g}_u(1,T), \overline{g}_u(X_1,T), \dots, \overline{g}_u(X_k,T)\}.$$

When a separating element is known, the associated rational univariate representation can easily be computed from the following traces:

$$\mathrm{Trace}(u^i) \quad (i = 0, \dots, N),$$

where $N = \dim_K A$ and

$$\mathrm{Trace}(u^i X_j) \quad (i = 0, \dots, n \ , \ j = 1, \dots, k),$$

where $n = \#(\mathcal{Z}_{\overline{K}}(\mathcal{P})) = \mathrm{degree}(\overline{\chi}_u(T))$.

Recipe VII: Rational Univariate Representation Computation.

The input is a Gröbner basis $\mathcal{G} \subset K[\mathcal{X}]$ of a set of polynomials $\mathcal{P}$ with a finite number of zeros.

1. Compute a monomial basis of $A = K[X_1, \dots, X_k]/I(\mathcal{P})$, and let N be the dimension of the K-vector space A.
2. Compute the matrix TrM and deduce the number of distinct roots n of the system by computing its rank.
3. Choose any u in

$$\{X_1 + iX_2 + \cdots + i^{k-1}X_k \mid 0 \le i \le (k-1)\binom{n}{2}\}$$

   (one among them is a separating element of $\mathcal{Z}_{\overline{K}}(\mathcal{P})$).
4. Compute, for $m \in \{1, \dots, N\}$, the numbers $\mathrm{Trace}(u^m)$, and deduce the polynomial $\chi_u(T)$, using Newton's formula (3.1).
5. Compute $\overline{\chi}_u(T)$, and let n' be its degree. If $n' < n$ then go to Step 3 and try another candidate separating element u.
6. Compute, for $j \in \{1, \dots, k\}$ and $i \in \{0, \dots, n\}$, the numbers $\mathrm{Trace}(X_j u^i)$ and deduce the polynomials $g_u(X_j, T)$.

7. If $n' = n$, then u is a separating element and the zeros of the system are:

$$\left(\frac{g_u(X_1,t)}{g_u(1,t)}, \dots\dots\dots, \frac{g_u(X_k,t)}{g_u(1,t)} \right),$$

where t is a root of $\chi_u(T)$.

More efficient criteria for determining if u is a separating element can be used. For example an element $u \in A$ is separating if and only if the polynomials $h_i(u) = \overline{\chi}_u(u)X_i - g_u(X_i, u)$ $(i = 1, \dots, k)$ are in the radical of $I(\mathcal{P})$ (see [19, 28]), which can be checked using Theorem 3.7. Since most of the traces needed have been computed before (when using the algorithm above), this test appears to be very efficient in practice. In the case of systems with integer coefficients, a separating element can be computed using modular arithmetic (see [28]). In general and with probability 1, a randomly chosen u is a separating element.

Example 3.11. We present here a nontrivial example in order to compare the lexicographic Gröbner basis and the rational univariate representation. Here the first variable is separating and all roots are regular. The polynomial system chosen has been extracted from the PoSSo collection of polynomial systems of equations:

$$\begin{aligned} P_1 &= 2x^2 + 2y^2 + 2z^2 + t^2 - t \\ P_2 &= 2xy + 2yz + 2zt - z \\ P_3 &= 2xz + z^2 + 2yt - y \\ P_4 &= 2x + 2y + 2z + t - 1. \end{aligned}$$

This polynomial system of equations is named *Katsura3* after its proposer and its class has often been used as a test for the efficiency of algorithms dealing with Gröbner basis computations. A Gröbner basis with respect to the lexicographic ordering was first computed. The lexicographic Gröbner basis computed with $t > z > y > x$ is

$$\begin{aligned} p_t =& 5913075t - 159690237696x^7 + 4884038x - 5913075 + 275119624x^2 \\ & -838935856x^3 - 6475723368x^4 + 27439610544x^5 + 31246269696x^6 \\ p_z =& 1971025z - 97197721632x^7 - 1678512x^2 - 9158924x + 814792828x^3 \\ & -2760941496x^4 - 12121915032x^5 + 73975630752x^6 \\ p_y =& 5913075y + 371438283744x^7 + 30947828x - 2024910556x^3 \\ & -132524276x^2 + 11520686172x^4 + 22645939824x^5 - 237550027104x^6 \\ p_x =& 128304x^8 - 93312x^7 + 15552x^6 + 3144x^5 - 1120x^4 + 36x^3 + 15x^2 - x. \end{aligned}$$

We observe that the coefficients in the univariate polynomial p_x are smaller than in the other polynomials.

Since we are in the Shape Lemma case, the variable x is separating; here is the corresponding rational univariate representation:

$$\begin{aligned}
\chi_x(T) &= 128304T^8 - 93312T^7 + 15552T^6 + 3144T^5 - 1120T^4 + 36T^3 \\
&\quad +15T^2 - T \\
g_x(1,T) &= 7185024T^7 - 4572288T^6 + 653184T^5 + 110040T^4 - 31360T^3 \\
&\quad +756T^2 + 210T - 7 \\
g_x(t,T) &= 3872448T^7 - 2607552T^6 + 408528T^5 + 63088T^4 - 20224T^3 \\
&\quad +540T^2 + 172T - 7 \\
g_x(z,T) &= 3032 64T^7 - 314928T^6 + 113544T^5 - 9840T^4 - 3000T^3 \\
&\quad +564T^2 - 12T \\
g_x(y,T) &= 699840T^7 - 449712T^6 + 74808T^5 + 1956T^4 - 1308T^3 \\
&\quad +174T^2 - 18x.
\end{aligned}$$

As mentioned in Proposition 3.10, $\chi_x = p_x$. The lexicographic Gröbner basis contains coefficients bigger than 10^{12} while the biggest coefficient of the rational univariate representation is smaller than 10^8. This behaviour, observed in [2], is due to the inversion of $g_x(1,T)$ (see Proposition 3.10).

3.3.2 Splitting the Rational Univariate Representation. In this part we suppose that

$$\{\chi_u(T), g_u(1,T), g_u(X_1,T), \ldots, g_u(X_k,T)\}$$

is a rational univariate representation for the elements of the finite set $\mathcal{Z}_{\overline{K}}(\mathcal{P})$.

The main advantage of the rational univariate representation is that we can apply many methods dealing with univariate polynomials in order to study the system. In order to simplify the output, one can for example make a squarefree decomposition, or even factorize the first polynomial of the rational univariate representation

$$\chi_u(T) = \prod_{i=1}^{k} \chi_{u,i}(T)$$

and also provide a representation of all the roots by a set of rational univariate representations:

$$\bigcup_{i=1}^{k} \{\chi_{u,i}(T), g_{u,i}(1,T), g_{u,i}(X_1,T), \ldots, g_{u,i}(X_k,T)\},$$

where $g_{u,i}(1,T) = g_u(1,T) \bmod \chi_{u,i}(T)$.

Example 3.12. Consider the following system where none of the variables is separating

$$\begin{aligned}
24tz - t^2 - z^2 - t^2z^2 - 13 &= 0 \\
24yz - y^2 - z^2 - y^2z^2 - 13 &= 0 \\
24ty - t^2 - y^2 - t^2y^2 - 13 &= 0.
\end{aligned}$$

A rational univariate representation is given by:

$$\begin{aligned}\chi_u(T) = {} & T^{16} - 5656T^{14} + 12508972T^{12} - 14213402440T^{10} + 9020869309270T^8 \\ & -3216081009505000T^6 + 6068330147542307 32T^4 \\ & -5131629663085504 4152T^2 + 10681305512246726 24689\end{aligned}$$

$$\begin{aligned}g_u(1,T) = {} & T^{15} - 4949T^{13} + 9381729T^{11} - 8883376525T^9 + 4510434654635T^7 \\ & -1206030378564375T^5 + 151708253688557683T^3 \\ & -641453707885688 0519T\end{aligned}$$

$$\begin{aligned}g_u(t,T) = {} & 71T^{14} - 355135T^{12} + 673508751T^{10} - 6332143597 91T^8 \\ & +314815356659869T^6 - 796776387004417 17T^4 \\ & +8618491509948092045T^2 - 2059560892895360 14429\end{aligned}$$

$$\begin{aligned}g_u(y,T) = {} & 86T^{14} - 418870T^{12} + 759804846T^{10} - 670485664238T^8 \\ & +307445009725282T^6 - 710124023665797 78T^4 \\ & +7099657810552674458T^2 - 1681909962025665 63226\end{aligned}$$

$$\begin{aligned}g_u(z,T) = {} & 116x^{14} - 483592T^{12} + 784226868T^{10} - 634062241592T^8 \\ & +270086313707548T^6 - 58355579408017944T^4 \\ & +5520988105236180668T^2 - 131448117382500870952.\end{aligned}$$

Noticing that $\chi_u(T)$ equals

$$\begin{aligned}&(T^4 - 1222T^2 + 371293) \cdot (T^4 - 1030T^2 + 190333) \cdot \\ &\qquad \cdot (T^4 - 2326T^2 + 484237) \cdot (T^4 - 1078T^2 + 31213),\end{aligned}$$

we can split the rational univariate representation in four components. For example, the component corresponding to the first factor is:

$$\begin{aligned}\chi_{u,1}(T) &= T^4 - 1222T^2 + 371293 \\ g_{u,1}(1,T) &= -1528597T^3 + 939034343T \\ g_{u,1}(t,T) &= 67229849947 - 104420381T^2 \\ g_{u,1}(y,T) &= 115704058093 - 203404643T^2 \\ g_{u,1}(z,T) &= 67229849947 - 104420381T^2.\end{aligned}$$

One advantage of the rational univariate representation is that it helps to keep track of root multiplicities. The polynomials of the rational univariate representation give an easy way to express the multiplicity of each root:

$$\forall\, x \in \mathcal{Z}_{\overline{K}}(\mathcal{P}), \quad \mu(x) = \frac{g_u(1, u(x))}{\overline{\chi}'_u(u(x))}.$$

where the prime on $\overline{\chi}_u$ denotes the derivative. Using this formula, we obtain the squarefree factorization of $\overline{\chi}_u(T)$ by computing the gcd's:

$$\chi_{u,i}(T) = \gcd(g_u(1,T) - i\overline{\chi}'_u(T), \overline{\chi}_u(T)), \quad i = 1, \ldots, \deg(\chi_u(T));$$

and we can compute the number of roots of given multiplicity i as the degree of $\overline{\chi}_{u,i}(T)$.

As a direct consequence of these last results, we can define the rational univariate representation of the roots of multiplicity i of $\mathcal{Z}_{\overline{K}(\mathcal{P})}$, which allows us to split the rational univariate representation without factorization (factorization can be very costly in practice):

$$\{\chi_{u,i}(T), g_{u,i}(1,T), g_{u,i}(X_1,T), \ldots, g_{u,i}(X_k,T)\},$$

where

$$g_{u,i}(1,T) = g_u(1,T) \bmod \overline{\chi}_{u,i}(T).$$

Example 3.13. Consider the following system

$$\begin{aligned}&24 - 92a - 92b - 113b^3 + 49a^4 + 49b^4 - 11a^5 - 11b^5 + a^6 + b^6 + 142a^2 + 284ab\\&\quad +142b^2 - 339a^2b - 339ab^2 + 294a^2b^2 + 196ab^3 - 55a^4b - 110a^3b^2 - 110a^2b^3\\&\quad -55ab^4 + 6a^5b + 15a^4b^2 + 20a^3b^3 + 15a^2b^4 + 6ab^5 - 113a^3 + 196a^3b\end{aligned}$$

$$c^3 + 3bc^2 + 3b^2c + b^3$$

$$8b^3 + 12b^2c - 12ab^2 + 6bc^2 - 12cab + 6a^2b + c^3 - 3ac^2 + 3a^2c - a^3.$$

A rational univariate representation is given by:

$$\begin{aligned}\chi_{u,1}(T) &= 8T^6 - 44T^5 + 98T^4 - 113T^3 + 71T^2 - 23T + 3\\ g_u(1,T) &= 24T^2 - 50T + 23\\ g_u(a,T) &= 24T^3 - 50T^2 + 23T\\ g_u(b,T) &= 22T^2 - 43T + 18\\ g_u(c,T) &= -22T^2 + 43T - 18.\end{aligned}$$

Since $\overline{\chi}_{u,1}(T) = 2T - 3$, $\overline{\chi}_{u,2} = 2T - 1$ and $\overline{\chi}_{u,3} = T - 1$, there is one root of multiplicity 1, one root of multiplicity 2, and one root of multiplicity 3. The rational univariate representations with respect to these multiplicities are:

$$\begin{aligned}&\overline{\chi}_{u,3} = T - 1,\ g_{u,3}(1,T) = g_{u,3}(a,T) = g_{u,3}(b,T) = -3,\ g_{u,3}(c,T) = 3\\&\overline{\chi}_{u,2} = 2T - 1,\ g_{u,2}(1,T) = 4,\ g_{u,2}(a,T) = g_{u,2}(b,T) = 2,\ g_{u,2}(c,T) = -2\\&\overline{\chi}_{u,1} = 2T - 3,\ g_{u,1}(1,T) = 2,\ g_{u,1}(a,T) = g_{u,1}(b,T) = 3,\ g_{u,1}(c,T) = -3.\end{aligned}$$

4. As Many Equations as Variables

We now consider systems of polynomial equations with a finite number of solutions defined by as many equations as variables. These polynomial systems are called *complete intersection polynomial systems.*

4.1 Generalities on Complete Intersection Polynomial Systems

Let $\{P_1, \ldots, P_k\}$ be a complete intersection polynomial system and A the corresponding quotient ring. We say that a linear form λ (i.e., a linear mapping from A to K) is *dualizing* if the bilinear form

$$\Lambda(a,b) = \lambda(ab)$$

is non-degenerate.

An important property of complete intersection polynomial systems is the following one.

Proposition 4.1. *There exists a dualizing linear form ℓ on A. For every dualizing form, the mapping associating to a in A the linear form $b \mapsto \ell(ab)$ is a one to one correspondence between elements of A and linear forms over A.*

The proof is far from obvious and can be found for example in [6]. In what follows, we describe explicitly how to obtain a dualizing form. Let $\mathcal{Y} = Y_1, \ldots, Y_k$ be indeterminates, just like $\mathcal{X}$. We define

$$B(X_1, \ldots, Y_k) := \det(P_{ij}) \in K[\mathcal{X}, \mathcal{Y}],$$

with P_{ij} equal to

$$\frac{P_i(Y_1, \ldots, Y_{j-1}, X_j, X_{j+1}, \ldots, X_k) - P_i(Y_1, \ldots, Y_{j-1}, Y_j, X_{j+1}, \ldots, X_k)}{X_j - Y_j}.$$

Observe that $A \otimes A$ can be viewed as the quotient ring of $K[\mathcal{X}, \mathcal{Y}]$ by the ideal generated by all $P(X_1, \ldots, X_k)$, $P(Y_1, \ldots, Y_k)$ for $P \in \mathcal{P}$. Now the class of $B(X_1, \ldots, Y_k)$ in $A \otimes A$ is called the *Bezoutian* of $P_1, \ldots, P_k$ and is denoted by $\mathrm{Bez}(x, y)$.

If $e_1, \ldots, e_N$ is a basis of A, an element a of $A \otimes A$ can be written as $\sum a_{i,j} e_i \otimes e_j$ and a linear form λ on A assigns to a, in a natural way, the element a^λ of A, defined as

$$a^\lambda = \sum_{i,j} a_{i,j} \lambda(e_i) e_j.$$

To a linear form λ we associate the element $\mathrm{Bez}(x, y)^\lambda$ of A where x and y denote the class in A of the variables in $\mathcal{X}$ and $\mathcal{Y}$, respectively. This mapping is surjective and the linear form ℓ associated to 1 is dualizing. It is called the *Kronecker symbol* (or global residue), and it is defined by $\mathrm{Bez}(x, y)^\ell = 1$. Moreover the element of A corresponding to a linear form λ in the bijection of Proposition 4.1 is $\mathrm{Bez}(x, y)^\lambda$ and $\ell(\mathrm{Bez}(x, y)^\lambda b) = \lambda(b)$.

In the univariate case, i.e., $k = 1$, denoting by H_i the Horner polynomial of degree i associated to $P = P_1$,

$$B(X, Y) = \frac{P(X) - P(Y)}{X - Y} = \sum_i H_{d-1-i}(Y) X^i = \sum_i Y^i H_{d-1-i}(X),$$

we obtain that the polynomial associated to λ is $\sum \lambda(H_{d-1-i}(Y)) X^i$. The Kronecker symbol ℓ sends $1, \ldots, X^{d-2}$ to 0 and X^{d-1} to 1.

Coming back to the multivariate case, the classical *Jacobian* of $P_1, \ldots, P_k$ agrees with $\mathrm{Bez}(x, x)$. The linear form associated to Jac in the correspondence of Proposition 4.1 is the Trace morphism and $\ell(\mathrm{Jac}\ u) = \mathrm{Trace}(u)$. As a consequence, the quadratic form Q_h can be presented as:

$$f \mapsto \ell(\mathrm{Jac}\ h f^2).$$

When all the zeros of $\{P_1, \ldots, P_k\}$ are regular, we have

$$\ell(f) = \sum_{x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})} \frac{f(x)}{\mathrm{Jac}(x)}.$$

One important property of complete intersection algebras is that the eigenspaces for the multiplication by f, which correspond as described above to the eigenvalues $f(x)$, are of dimension 1: eigenvectors are $\mathrm{Bez}(X, x) \in \overline{A}$ with $x \in \mathcal{Z}_{\overline{K}}(\mathcal{P})$. This gives fixed point properties and interesting numerical methods for solving polynomial systems [3, 10].

4.2 Recipes for Polynomial System Solving When the Number of Equations Equals the Number of Unknowns

Recipe VIII: Number of Different Solutions.

- Start from a basis $\mathcal{A}$ of A as a K-vector space (for example, given by the monomials under the staircase of a Gröbner basis).
- Compute the Jacobian determinant of $P_1, P_2, \ldots, P_k$:

$$\mathrm{Jac} = \begin{vmatrix} \frac{\partial P_1}{\partial X_1} & \cdots & \frac{\partial P_1}{\partial X_k} \\ \vdots & & \vdots \\ \frac{\partial P_k}{\partial X_1} & \cdots & \frac{\partial P_k}{\partial X_k} \end{vmatrix}.$$

- Compute the matrix $\mathcal{L}_{\mathrm{Jac}}$ (i.e., the matrix of the endomorphism L_{Jac} with respect to $\mathcal{A}$).
- The rank of the matrix $\mathcal{L}_{\mathrm{Jac}}$ is equal to number of different solutions of the polynomial system.

Example 4.2 (Continuation of Example 2.9). The next excerpts of a *Maple* session show the computation of the matrix $\mathcal{L}$ for the polynomial system of equations in Example 2.9.

```
Jac := det(jacobian(F,[x,y]));
```

$$Jac := -4xy^2 + x^2y - 8x^2 - 4y^2 - 4x^3$$

```
Abasis := [1,y,y**2,y**3,x,x*y]:
for omega in Abasis do normalf(omega*Jac,GB,[x,y],tdeg) od;
```

$$-6y^3 - 26xy - 8y^2$$
$$2y^3 - 8xy$$
$$-20y^3 - 64xy$$
$$56y^3 + 64xy$$
$$20xy + 4y^3$$
$$-4y^3 + 16xy$$

Now in the matrix $P = \mathcal{L}_{\mathrm{Jac}}$ the i-th column consists of the coefficients of the i-th polynomial in the above list with respect to the given monomial basis of A:

$$P := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 & 0 & 0 \\ -6 & 2 & -20 & 56 & 4 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -26 & -8 & -64 & 64 & 20 & 16 \end{pmatrix}$$

```
rank(P);
```

$$3$$

Thus the considered polynomial system of equations has three different solutions.

5. Gröbner Bases and Numerical Approximations

Once a Gröbner basis is known, a procedure for determining the solutions numerically is as follows. According to Stickelberger's Theorem 3.3, the k matrices $\mathcal{L}_{X_1}, \ldots, \mathcal{L}_{X_k}$ have the following property: for each i, the set of i-th coordinates of the common zeros of $\mathcal{P}$ coincides with the set of eigenvalues of $\mathcal{L}_{X_i}$. The main problem is to recombine the different eigenvalues in order to get the solutions of the polynomial system.

When all the solutions are regular, the idempotents e_x associated to $x \in \mathcal{Z}_{\bar{K}}(\mathcal{P})$ are, according to Stickelberger's Theorem 3.3, the eigenvectors of the various $\mathcal{L}_{X_i}$'s. So, all matrices $\mathcal{L}_{X_i}$, $i = 1, \ldots, k$, are diagonalizable in the same basis of eigenvectors. If, moreover, the first variable is separating (this means that the characteristic polynomial of $\mathcal{L}_{X_1}$ is squarefree), we can compute the eigenvalues and eigenvectors of $\mathcal{L}_{X_1}$. The eigenvalues are the first coordinates of the elements $x = (x_1, \ldots, x_k) \in \mathcal{Z}_{\bar{K}}(\mathcal{P})$, and the eigenvectors f_x obtained are proportional to the idempotents e_x. Now

$$\mathcal{L}_{X_1} \cdot f_x = x_1 f_x, \;\; \mathcal{L}_{X_2} \cdot f_x = x_2 f_x, \;\; \ldots, \;\; \mathcal{L}_{X_k} \cdot f_x = x_k f_x,$$

so that $x_2, \ldots, x_k$ can be computed from f_x and $\mathcal{L}_{X_2}, \ldots, \mathcal{L}_{X_k}$.

In the general case, we can use for example the following lemma [20].

Lemma 5.1. *If $\{A_1, \ldots, A_n\}$ is a commuting family of matrices ($A_i A_j = A_j A_i$ for every pair from the family) then there exists a unitary matrix U such that*

$$T_i = U^* A_i U$$

is upper triangular.

Applying this lemma to the commuting family of matrices $\{\mathcal{L}_{X_1},\dots,\mathcal{L}_{X_k}\}$, it is possible to read off from the diagonal of the upper triangular matrices $T_i=U^*\mathcal{L}_{X_i}U$ the coordinates of the solutions of the polynomial system. The construction of the matrix U is not difficult in the exact arithmetic/data case but there are interesting problems in the floating point/inexact data case [11].

`Recipe IX`: Approximating the Solutions (Regular Case)

- Compute the matrices $\mathcal{L}_{X_1},\dots,\mathcal{L}_{X_k}$.
- Compute the characteristic polynomial of $\mathcal{L}_{X_1}$ and check if it is squarefree. If it is not squarefree, go to `Recipe X`.
- If it is squarefree compute the eigenvalues of $\mathcal{L}_{X_1}$, using any numerical algorithm.
- Compute the eigenvectors of $\mathcal{L}_{X_1}$ using any numerical algorithm and compute the other coordinates of the solutions as indicated above.

`Recipe X`: Approximating the Solutions

- Compute the matrices $\mathcal{L}_{X_1},\dots,\mathcal{L}_{X_k}$.
- Compute the eigenvalues of $\mathcal{L}_{X_1},\dots,\mathcal{L}_{X_k}$ using any numerical algorithm.
- Combine the obtained results to get the solutions by using, for example, the preceding lemma.

Example 5.2 (End of Example 2.8). Finally the matrices $\mathcal{L}_x$ and $\mathcal{L}_y$, together with their eigenvalues, are computed.

```
for omega in Abasis do normalf(omega*x,GB,[x,y],tdeg) od;
```

$$x$$

$$xy$$

$$4xy+y^3$$

$$-2y^3$$

$$-1/2xy+1/2y^2$$

$$-2xy$$

```
for omega in Abasis do normalf(omega*y,GB,[x,y],tdeg) od;
```

$$y$$

$$y^2$$

$$y^3$$

$$-6y^3-16xy$$

$$xy$$

$$4xy+y^3$$

Similarly to $\mathcal{L}_{\mathrm{Jac}}$, these lists give us $M_1 = \mathcal{L}_x$ and $M_2 = \mathcal{L}_y$:

$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & -1/2 & -2 \end{pmatrix} \quad M_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -6 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -16 & 1 & 4 \end{pmatrix}$$

```
eigenvals(M1);eigenvals(M2);
```

$$0, 0, 0, 0, -2, -2$$
$$0, 0, 0, 0, 2, -4$$

```
subs(x=-2,y=2,F),subs(x=-2,y=-4,F),subs(x=0,y=0,F);
```

$$[0, 0], [0, 0], [0, 0]$$

So, the three solutions of the polynomial system of equations we consider are $(-2, 2)$, $(-2, -4)$ and $(0, 0)$.

References

1. W. W. Adams and P. Loustaunau (1994): *An Introduction to Gröbner Bases.* Graduate Studies in Mathematics **3**, Amer. Math. Soc.
2. M.-E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann (1996): *Zeros, Multiplicities and Idempotents for Zerodimensional Systems.* Algorithms in Algebraic Geometry and Applications, Progress in Mathematics **143**, 1–20, Birkhäuser.
3. W. Auzinger and H. J. Stetter (1988): *An Elimination Algorithm for the Computation of all Zeros of a System of Multivariate Polynomial Equations.* Int. Series in Numerical Mathematics **86**, 11–30, Birkhäuser.
4. E. Becker (June 1996): Private communication.
5. E. Becker, M. G. Marinari, T. Mora, and C. Traverso (1993): *The Shape of the Shape Lemma.* Proceedings of ISSAC–94, 129–133, ACM Press.
6. E. Becker, J.-P. Cardinal, M.-F. Roy, and Z. Szafraniec (1996): *Multivariate Bezoutians Kronecker Symbol and Eisenbud-Levine formula.* Algorithms in Algebraic Geometry and Applications, Progress in Mathematics **143**, 79–104, Birkhäuser.
7. E. Becker and T. Wörmann (1996): *Radical computation of a zero-dimensional ideal ad real root counting.* Mathematics and Computers in Simulation **42** (4–6), 561–569.
8. T. Becker and V. Weispfenning (1993): *Groebner Bases, a Computational Approach to Commutative Algebra.* Graduate Texts in Mathematics **141**. Springer-Verlag, New York Berlin Heidelberg.
9. B. Buchberger (1985): *Gröbner bases: an algorithmic method in polynomial ideal theory.* Multidimensional Systems Theory (N. K. Bose Ed.), Chapter 6, 184–232, Reidel Publishing Company, Dordrecht.

10. J.-P. Cardinal (1993): *Dualité et algorithmes itératifs pour la solution des systèmes polynomiaux.* Thèse, Université de Rennes I.
11. R.M. Corless (1996): *Groebner bases and matrix eigenproblems.* SIGSAM Bulletin **30** (4), 26–32.
12. D. Cox, J. Little, and D. O'Shea (1993): *Ideals, Varieties and Algorithms.* Undergraduate Texts in Mathematics. Springer-Verlag, New York Berlin Heidelberg.
13. D. Eisenbud (1995): *Commutative Algebra with a View Toward Algebraic Geometry.* Graduate Texts in Mathematics **150**, Springer-Verlag, New York Berlin Heidelberg.
14. J.C. Faugère, P. Gianni, D. Lazard, and T. Mora (1994): *Efficient computation of zero-dimensional Gröbner bases by change of ordering.* Journal of Symbolic Computation **16** (4), 329–344.
15. P. Gianni (1987): *Properties of Gröbner bases under specialization.* Proceedings Eurocal–87. Lectures Notes in Computer Science **378**, 293–297, Springer-Verlag, Berlin Heidelberg New York.
16. P. Gianni and T. Mora (1989): *Algebraic solution of polynomial equations using Gröbner bases.* Proceedings AAECC–5. Lectures Notes in Computer Science **359**, 247–257, Springer-Verlag, Berlin Heidelberg New York.
17. P. Gianni, V. Miller, and B. Trager (1988): *Decomposition of algebras.* Lecture Notes in Computer Science **356**, 300–308, Springer-Verlag, Berlin Heidelberg New York.
18. M. Giusti and J. Heintz (1993): *La determination des points isoles et de la dimension d'une variete algebrique peut se faire en temps polynomial.* Computational Algebraic Geometry and Commutative Algebra, Symposia Mathematica, vol. XXXIV, 216–256, Cambridge University Press.
19. L. González-Vega and G. Trujillo (1995): *Using symmetric functions to describe the solution set of a zero dimensional ideal.* Lecture Notes in Computer Science **948**, 232–247, Springer-Verlag, Berlin Heidelberg New York.
20. R. Horn and C. Johnson (1985): *Matrix Analysis.* Cambridge University Press.
21. M. Kalkbrener (1987): *Solving systems of algebraic equations by using Gröbner bases.* Proceedings Eurocal–87. Lectures Notes in Computer Science **378**, 282–292, Springer-Verlag, Berlin Heidelberg New York.
22. T. Krick and L.M. Pardo (1996): *A computational method for diophantine approximation.* Algorithms in Algebraic Geometry and Applications, Progress in Mathematics **143**, 193–254, Birkhäuser, Basel.
23. Y.N. Lakshman and D. Lazard (1991): *On the complexity of zero-dimensional algebraic systems.* Effective Methods in Algebraic Geometry, Progress in Mathematics **94**, 217–225, Birkhäuser, Basel.
24. B. Mishra (1993): *Algorithmic Algebra.* Texts and Monographs in Computer Science. Springer-Verlag, Berlin Heidelberg New York.
25. M. Moreno-Maza (1997): *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Équations Algébriques.* Doctoral Thesis, Université Paris 6.
26. H.M. Möller (1993): *Systems of algebraic equations solved by means of endomorphisms.* Applied Algebra and Error Correcting Codes, Lecture Notes in Computer Science **673**, 43–56, Springer-Verlag, Berlin Heidelberg New York.
27. P. Pedersen, M.-F. Roy, and A. Szpirglas (1993): *Counting real zeros in the multivariate case.* Computational Algebraic Geometry, Progress in Mathematics **109**, 61–76, Birkhäuser, Basel.
28. F. Rouillier (1996): *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux.* Doctoral Thesis, Université de Rennes I.
29. B.L. Van der Waerden (1950): *Modern Algebra II.* F. Ungar Publishing Co.

30. V. Weispfenning (1995): *Solving parametric polynomial equations and inequalities by symbolic algorithms.* Computer Algebra in Science and Engineering, 163–179, World Scientific, Singapore.

31. K. Yokoyama, M. Noro, and T. Takeshima (1992): *Solutions of systems of algebraic equations and linear maps on residue class rings.* Journal of Symbolic Computation **14**, 399–417.

Chapter 3. Lattice Reduction

Frits Beukers

1. Introduction

We shall give an introduction to the LLL-algorithm over $\mathbb{Z}$. The algorithm is due to L. Lovász, H.W. Lenstra and A.K. Lenstra. It is concerned with the problem of finding a shortest nonzero vector in a lattice. In Section 2, we begin by introducing the relevant background material on lattices. Then we proceed to describe lattice reduction and finding shortest nonzero vectors in dimension 2. Section 4 presents the core result of this chapter: LLL-lattice reduction in any dimension. Section 5 deals with the implementation of the LLL-algorithm, and the last section discusses an application of the algorithm to the problem of finding $\mathbb{Z}$-linear combinations of a given set of real numbers with small values.

2. Lattices

Consider $\mathbb{R}^n$ with the standard inner product, which we denote by $\mathbf{v} \cdot \mathbf{w}$.

Lemma 2.1. *Let G be an additive subgroup of $\mathbb{R}^n$. Then G is discrete in $\mathbb{R}^n$ if and only if there exist $\mathbb{R}$-linearly independent elements $\mathbf{v}_1, \ldots, \mathbf{v}_r \in G$ such that $G = \{x_1\mathbf{v}_1 + \cdots + x_r\mathbf{v}_r \mid x_1, \ldots, x_r \in \mathbb{Z}\}$.*

Proof. Suppose $G = \{x_1\mathbf{v}_1 + \cdots + x_r\mathbf{v}_r \mid x_1, \ldots, x_r \in \mathbb{Z}\}$ with $\mathbf{v}_1, \ldots, \mathbf{v}_r$ linearly independent over $\mathbb{R}$. Let μ be the minimum of $|x_1\mathbf{v}_1 + \cdots + x_r\mathbf{v}_r|$ as $x_1, \ldots, x_r$ run over all real numbers such that $x_1^2 + \cdots + x_r^2 = 1$. Since the $\mathbf{v}_i$ are independent, this minimum is nonzero. Hence we have, for any $x_1, \ldots, x_r \in \mathbb{R}$, that

$$|x_1\mathbf{v}_1 + \cdots + x_r\mathbf{v}_r| \geq \mu\sqrt{x_1^2 + \cdots + x_r^2}.$$

Hence, for any nonzero $\mathbf{v} \in G$, we obtain $|\mathbf{v}| \geq \mu$. Hence G is discrete.

Suppose, conversely, that G is discrete. Let r be the dimension of the $\mathbb{R}$-linear span of G and choose r linearly independent (over $\mathbb{R}$) elements $\mathbf{w}_1, \ldots, \mathbf{w}_r$ of G. Consider the set

$$F = \{\mathbf{x} \in G \mid \mathbf{x} = \mu_1\mathbf{w}_1 + \cdots + \mu_r\mathbf{w}_r,\ \forall i : 0 \leq \mu_i \leq 1\}.$$

Since G is discrete, the set F is finite. For each $i = 1, \ldots, r$ we choose $\mathbf{v}_i \in F$ such that $\mathbf{v}_i = \mu_i\mathbf{w}_i + \cdots + \mu_r\mathbf{w}_r$ with $\mu_i > 0$ and minimal. Since $\mathbf{w}_i \in F$, such an element always exists. Clearly the $\mathbf{v}_i$ are also $\mathbb{R}$-linearly independent. Let $\mathbf{v} \in G$ and write $\mathbf{v} = \sum_{i=1}^{r} \lambda_i \mathbf{v}_i$. For each i, let ν_i be equal to λ_i minus its largest integral part. Then $\mathbf{v}' := \sum_{i=1}^{r} \nu_i \mathbf{v}_i$ is also an element of G. We assert that $\nu_i = 0$ for all i. Suppose not, then choose j minimal such that $\nu_j > 0$. Then $\mathbf{v}'$ written with respect to the $\mathbf{w}_i$ looks like $\mathbf{v}' = \nu_j \mu_j \mathbf{w}_j + \cdots$, contradicting the minimality in our choice of $\mathbf{v}_j$.

Definition 2.2. A *lattice* in $\mathbb{R}^n$ is a discrete subgroup of the additive group $\mathbb{R}^n$.

A set of independent generators of a lattice L is called a *(lattice) basis.* The *rank* of a lattice L is the usual one in the sense of linear algebra and it equals the number of elements of a lattice basis.

Lemma 2.3. *Let $\mathbf{w}_1, \ldots, \mathbf{w}_r$ and $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be any two bases of a lattice L. Then there exists an $r \times r$-matrix M with integral entries and* $\det(M) = \pm 1$ *such that $\mathbf{w}_i = M\mathbf{v}_i$ for $i = 1, \ldots, r$.*

Proof. Since $\{\mathbf{v}_i\}_i$ and $\{\mathbf{w}_i\}_i$ are bases of L, there exist $r \times r$-matrices M, N with integral entries such that $\mathbf{w}_i = M\mathbf{v}_i$ and $\mathbf{v}_i = N\mathbf{w}_i$ for all i. Hence $MN = \mathrm{Id}_r$ and $\det(M)\det(N) = 1$. Since both determinants are integers, we conclude that $\det(M) = \det(N) = \pm 1$.

Definition 2.4. Let L be a lattice in $\mathbb{R}^n$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be a basis of L. Then we define the *determinant* of a lattice by $\sqrt{\det(\mathbf{v}_i \cdot \mathbf{v}_j)}$. Notation: $d(L)$.

The matrix $(\mathbf{v}_i \cdot \mathbf{v}_j)_{i,j=1,\ldots,r}$ is called the *Gram-matrix* of $\mathbf{v}_1, \ldots, \mathbf{v}_r$. Note that the Gram-matrix is symmetric with positive eigenvalues. Hence its determinant is positive and we can take its square root.

For future use we make the following observation. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be a basis of $\mathbb{R}^n$ and let V be the matrix whose columns consist of these vectors. Then $V^\top V$ is the matrix whose entries consist of the inner products $\mathbf{v}_i \cdot \mathbf{v}_j$. Hence $d(L)^2 = \det(\mathbf{v}_i \cdot \mathbf{v}_j)_{i,j=1,\ldots,n} = \det(V^\top)\det(V)$. So we conclude that $d(L) = |\det(\mathbf{v}_1, \ldots, \mathbf{v}_r)|$.

From the theory of lattices we have the following theorem.

Theorem 2.5 (Minkowski). *Let L be a lattice of rank r. Let $\mathbf{v}_1 \in L$ be a shortest nonzero vector, $\mathbf{v}_2 \in L$ a shortest vector independent of $\mathbf{v}_1$, etc., and let finally $\mathbf{v}_r \in L$ be a shortest vector independent of $\mathbf{v}_1, \ldots, \mathbf{v}_{r-1}$. Then,*

$$|\mathbf{v}_1| \cdots |\mathbf{v}_r| \leq \frac{2^r}{\mathrm{vol}(B_r)} d(L)$$

where $\mathrm{vol}(B_r)$ *is the volume of the unit ball in* $\mathbb{R}^r$.

We add that $\mathrm{Vol}(B_r) = \pi^{r/2}/\Gamma(1+r/2)$. As B_r contains the (hyper-)cube whose vertices all have coordinates $\pm 1/\sqrt{r}$ we obtain $\mathrm{Vol}(B_r) \geq (2/\sqrt{r})^r$. As a consequence we find that $|\mathbf{v}_1| \leq \sqrt{r}\, d(L)^{1/r}$.

An important problem with many applications is the following one.

Problem 2.1. Given a basis of a lattice L. Determine a nonzero vector in L with minimal length.

A possible application would be the determination of $a, b \in \mathbb{Z}$ such that $p = a^2 + b^2$ for a prime p with $p \equiv 1 \bmod 4$.

Example 2.6. Let p be a prime with $p \equiv 1 \bmod 4$. Find $z \in \mathbb{Z}$ such that $z^2 \equiv -1 \bmod p$ (there exist algorithms to do this quickly). Consider the lattice

$$L = \{(x, y) \in \mathbb{Z}^2 \mid x \equiv zy \bmod p\}.$$

Check that $(p, 0)$ and $(z, 1)$ form a basis of L. Hence $d(L) = p$. Denote a shortest nonzero vector in L by (a, b). By Minkowski's theorem we have that $|(a,b)|^2 \leq 4p/\pi$, hence $a^2+b^2 < 2p$. On the other hand, $a^2+b^2 \equiv (bz)^2+b^2 \equiv 0 \bmod p$. Hence p divides a^2+b^2 and we conclude that $p = a^2 + b^2$.

Exercise 2.7. Let L be a lattice with base $\mathbf{b}_1, \ldots, \mathbf{b}_r$. Show that the length of every nonzero vector in L is bounded below by the smallest eigenvalue of the Gram-matrix of the $\mathbf{b}_i$.

3. Lattice Reduction in Dimension 2

In case $n = 2$ there is a very efficient algorithm to find shortest vectors in lattices. Let L be a lattice in $\mathbb{R}^2$ with basis $\mathbf{v}_1, \mathbf{v}_2$ and assume $|\mathbf{v}_2| \geq |\mathbf{v}_1|$.

Algorithm 3.1 (Euclid).

```
loop:
choose $k \in \mathbb{Z}$ such that $-\frac{1}{2}\mathbf{v}_1 \cdot \mathbf{v}_1 < (\mathbf{v}_2 - k\mathbf{v}_1) \cdot \mathbf{v}_1 \leq \frac{1}{2}\mathbf{v}_1 \cdot \mathbf{v}_1$;
$\mathbf{v}_2 := \mathbf{v}_2 - k\mathbf{v}_1$;
if $|\mathbf{v}_2| \geq |\mathbf{v}_1|$ then stop;
else interchange $\mathbf{v}_1$ and $\mathbf{v}_2$; goto loop;
fi.
```

We assert that this algorithm terminates and that $\mathbf{v}_1$ is a shortest nonzero vector in L and $\mathbf{v}_2$ is a shortest vector in $L \setminus \{c\mathbf{v}_1 | \ c \in \mathbb{R}\}$.

Proof. First we show termination. At the start of every loop the vector $\mathbf{v}_1$ is strictly smaller than at the start of the previous loop. Since every bounded disc contains only finitely many elements of L (L is discrete), the algorithm terminates.

We now show correctness of our algorithm. Let $\mathbf{v}_1, \mathbf{v}_2$ be the result of the algorithm. In particular we have that $|\mathbf{v}_2 \cdot \mathbf{v}_1| \leq \frac{1}{2}|\mathbf{v}_1|^2$ and $|\mathbf{v}_2| \geq |\mathbf{v}_1|$. Choose any nonzero $\mathbf{v} \in L$. There exist $a, b \in \mathbb{Z}$ such that $\mathbf{v} = a\mathbf{v}_1 + b\mathbf{v}_2$. Notice that

$$\begin{aligned}|\mathbf{v}|^2 &= a^2|\mathbf{v}_1|^2 + 2ab(\mathbf{v}_1 \cdot \mathbf{v}_2) + b^2|\mathbf{v}_2|^2 \\ &\geq a^2|\mathbf{v}_1|^2 - |ab||\mathbf{v}_1|^2 + b^2|\mathbf{v}_2|^2 \\ &\geq (a^2 - |ab| + b^2)|\mathbf{v}_1|^2 \geq |\mathbf{v}_1|^2.\end{aligned}$$

Hence $\mathbf{v}_1$ is a shortest vector. Now suppose $\mathbf{v}$ independent of $\mathbf{v}_1$, i.e., $b \neq 0$. Then,

$$\begin{aligned}|\mathbf{v}|^2 &\geq a^2|\mathbf{v}_1|^2 - |ab||\mathbf{v}_1|^2 + \frac{1}{4}b^2|\mathbf{v}_1|^2 + \frac{3}{4}b^2|\mathbf{v}_2|^2 \\ &= (|a| - |b|/2)^2|\mathbf{v}_1|^2 + \frac{3}{4}b^2|\mathbf{v}_2|^2 \\ &\geq |\mathbf{v}_2|^2 \quad \text{if } |b| > 1.\end{aligned}$$

If $b = \pm 1$ then

$$|\mathbf{v}|^2 \geq a^2|\mathbf{v}_1|^2 - |a||\mathbf{v}_1|^2 + |\mathbf{v}_2|^2 \geq |\mathbf{v}_2|^2.$$

Hence $\mathbf{v}_2$ is a shortest vector independent of $\mathbf{v}_1$.

Lemma 3.2. *Let $\mathbf{v}_1$ be the result of the previous algorithm. Then $|\mathbf{v}_1| \leq (4/3)^{1/4}d(L)^{1/2}$.*

Proof. We have $|\mathbf{v}_2| \geq |\mathbf{v}_1|$ and $|\mathbf{v}_1 \cdot \mathbf{v}_2| \leq \frac{1}{2}|\mathbf{v}_1|^2$. Notice that

$$\begin{aligned}d(L)^2 &= \begin{vmatrix} |\mathbf{v}_1|^2 & \mathbf{v}_1 \cdot \mathbf{v}_2 \\ \mathbf{v}_1 \cdot \mathbf{v}_2 & |\mathbf{v}_2|^2 \end{vmatrix} \\ &= |\mathbf{v}_1|^2|\mathbf{v}_2|^2 - |\mathbf{v}_1 \cdot \mathbf{v}_2|^2 \\ &\geq |\mathbf{v}_1|^4 - \frac{1}{4}|\mathbf{v}_1|^4 = \frac{3}{4}|\mathbf{v}_1|^4.\end{aligned}$$

Taking fourth roots on both sides yields our inequality.

Notice that the inequality sign of the previous lemma becomes equality precisely when $|\mathbf{v}_1| = |\mathbf{v}_2|$ and $|\mathbf{v}_1 \cdot \mathbf{v}_2| = \frac{1}{2}|\mathbf{v}_1|^2$. This case corresponds to the hexagonal lattice.

The most important feature of Euclid's algorithm is its remarkably short runtime as shown by the following exercise.

Exercise 3.3. Let l be the length of the shortest nonzero vector in a lattice of rank 2. Let v_1, v_2 be the initial basis of the lattice. Prove that Euclid's algorithm ends in $O(\log(|v_1|/l))$ iterations.

Exercise 3.4. Let c be a real number in $]1/\sqrt{3}, 1[$. Suppose that we replace the stopping condition $|\mathbf{v}_2| \geq |\mathbf{v}_1|$ in our previous algorithm by $|\mathbf{v}_2| \geq c|\mathbf{v}_1|$. Let $\mathbf{v}_1$ be the result of our new algorithm. Show that for any nonzero $\mathbf{v} \in L$ we have $|\mathbf{v}| \geq c|\mathbf{v}_1|$.

4. Lattice Reduction in Any Dimension

In case $n \geq 3$ there are hardly any polynomial time, general purpose methods with a shortest lattice vector as guaranteed output. However, in 1982 L. Lovász, H.W. Lenstra and A.K. Lenstra (cf. [2]) proposed an algorithm which produces in polynomial time a lattice vector whose length is at most a known factor larger than the shortest possible length.

Before describing the algorithm we review the Gram-Schmidt orthogonalisation procedure. Let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be (not necessarily independent) vectors in $\mathbb{R}^n$. Define recursively,

$$\begin{aligned} \mathbf{v}_1^* &= \mathbf{v}_1 \\ \mathbf{v}_i^* &= \mathbf{v}_i - {\sum_{j<i}}' \frac{\mathbf{v}_i \cdot \mathbf{v}_j^*}{|\mathbf{v}_j^*|^2} \mathbf{v}_j^* \qquad (i = 2, \ldots, r) \end{aligned}$$

where the ' sign in the summation denotes deletion of terms where $\mathbf{v}_j^* = 0$. The vectors $\mathbf{v}_i^*$ consist of (possibly) some zero vectors and an orthogonal basis of the space spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_r$.

Notice that $|\mathbf{v}_k^*| \leq |\mathbf{v}_k|$ for all k and that $\det(\mathbf{v}_i \cdot \mathbf{v}_j) = \det(\mathbf{v}_i^* \cdot \mathbf{v}_j^*)$. Hence,

$$\det(\mathbf{v}_i \cdot \mathbf{v}_j) = \prod_{i=1}^{r} |\mathbf{v}_i^*|^2 \leq \prod_{i=1}^{r} |\mathbf{v}_i|^2.$$

This inequality is known as *Hadamard's inequality.* In particular, when $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is a basis of $\mathbb{R}^n$ we obtain

$$|\det(\mathbf{v}_1, \ldots, \mathbf{v}_n)| \leq \prod_{i=1}^{n} |\mathbf{v}_i|.$$

In the sequel, whenever we have a set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$, we denote by $\mathbf{v}_1^*, \ldots, \mathbf{v}_r^*$ the result of the Gram-Schmidt procedure. The so-called *Gram-Schmidt coefficients* $\mathbf{v}_i \cdot \mathbf{v}_j^* / |\mathbf{v}_j^*|^2$ are denoted by μ_{ij}. We take $\mu_{ij} = 0$ if $\mathbf{v}_j^* = 0$.

In practice we shall only be interested in the inner products $\mathbf{v}_i \cdot \mathbf{v}_j^*$. To compute these products we use the following algorithm.

Algorithm 4.1 (Gram-Schmidt).

```
$G := (\mathbf{v}_i \cdot \mathbf{v}_j)$;
for $i$ from 1 to $n$ do
  if $G_{ii} \neq 0$ then
    for $j$ from $i+1$ to $n$ do
      subtract $G_{ij}/G_{ii}$ times the $i$-th column from the $j$-th column;
    od;
  fi;
od.
```

When the algorithm terminates, the matrix G has the products $\mathbf{v}_i \cdot \mathbf{v}_j^*$ as entries.

We are now ready to discuss LLL-reduction.

Definition 4.2. Let L be a lattice. A basis $\mathbf{b}_1, \ldots, \mathbf{b}_r$ of L is called *LLL-reduced* if

$$|\mu_{ij}| \leq \frac{1}{2} \quad \text{whenever } 1 \leq j < i \leq r$$

and

$$|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*|^2 \geq \frac{3}{4}|\mathbf{b}_{i-1}^*|^2 \quad \text{whenever } 1 < i \leq r.$$

The second condition can be rewritten as $|\mathbf{b}_i^*|^2 \geq (\frac{3}{4} - \mu_{i,i-1}^2)|\mathbf{b}_{i-1}^*|^2$ and is known as Lovász's condition. The vector $b_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*$ can be interpreted as the projection of b_i on the orthogonal complement of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-2}$. In the special case $r = 2$ the conditions read $|\mathbf{b}_2 \cdot \mathbf{b}_1| \leq |\mathbf{b}_1|^2$ and $|\mathbf{b}_2|^2 \geq \frac{3}{4}|\mathbf{b}_1|^2$.

Theorem 4.3. *Let $\mathbf{b}_1, \ldots, \mathbf{b}_r$ be an LLL-reduced basis of a lattice L. Then,*

1. $d(L) \leq \prod_{i=1}^r |\mathbf{b}_i| \leq 2^{r(r-1)/4} d(L)$.
2. $|\mathbf{b}_1| \leq 2^{(r-1)/4} d(L)^{1/r}$
3. *For every nonzero* $\mathbf{x} \in L$ *we have* $|\mathbf{b}_1| \leq 2^{(r-1)/2}|\mathbf{x}|$.
4. *For any linearly independent set of vectors* $\mathbf{x}_1, \ldots, \mathbf{x}_t \in L$ *we have* $|\mathbf{b}_j| \leq 2^{(r-1)/2} \max(|\mathbf{x}_1|, \ldots, |\mathbf{x}_t|)$ *for* $1 \leq j \leq t$.

Proof. First note the following inequalities,

$$\begin{aligned} |\mathbf{b}_i|^2 &= |\mathbf{b}_i^*|^2 + \mu_{i,i-1}^2|\mathbf{b}_{i-1}^*|^2 + \cdots + \mu_{i,1}^2|\mathbf{b}_1^*|^2 \\ &\leq |\mathbf{b}_i^*|^2 + \frac{1}{4}|\mathbf{b}_{i-1}^*|^2 + \cdots + \frac{1}{4}|\mathbf{b}_1^*|^2. \end{aligned}$$

Furthermore, $|\mathbf{b}_j^*|^2 \geq \frac{1}{2}|\mathbf{b}_{j-1}^*|^2$ as a consequence of the Lovász condition. Hence $|\mathbf{b}_j^*|^2 \leq 2^{i-j}|\mathbf{b}_i^*|^2$ whenever $j \leq i$. Hence for all i we have

$$\begin{aligned} |\mathbf{b}_i|^2 &\leq \left[1 + \frac{1}{4}(2 + 2^2 + \cdots + 2^{i-1})\right] |\mathbf{b}_i^*|^2 \\ &= \frac{2^{i-1}+1}{2}|\mathbf{b}_i^*|^2 \leq 2^{i-1}|\mathbf{b}_i^*|^2. \end{aligned}$$

We are now ready to prove the statements of our theorem. First of all,

$$d(L) = \prod_{i=1}^r |\mathbf{b}_i^*| \leq \prod_{i=1}^r |\mathbf{b}_i| \leq 2^{r(r-1)/4} \prod_{i=1}^r |\mathbf{b}_i^*| = 2^{r(r-1)/4} d(L).$$

This proves part 1.

Secondly, whenever $1 \leq j < i \leq r$ we have

$$|\mathbf{b}_j| \leq 2^{(j-1)/2}|\mathbf{b}_j^*| \leq 2^{(i-j)/2}2^{(j-1)/2}|\mathbf{b}_i^*| = 2^{(i-1)/2}|\mathbf{b}_i^*|.$$

Application of the latter inequality to the case $j = 1$ yields

$$|\mathbf{b}_1|^r \le 2^{r(r-1)/4}|\mathbf{b}_1^*||\mathbf{b}_2^*|\cdots|\mathbf{b}_r^*| = 2^{r(r-1)/4}d(L).$$

Hence $|\mathbf{b}_1| \le 2^{(r-1)/4}d(L)^{1/r}$, which proves part 2.

Note that part 3 is a special case of 4 with $t = 1$.

For the proof of part 4 we choose k minimal such that $\mathbf{x}_1, \ldots, \mathbf{x}_t$ lie in the span of $\mathbf{b}_1, \ldots, \mathbf{b}_k$. Suppose $\mathbf{x}_i = \sum_{1\le j\le k} r_{ij}\mathbf{b}_j = \sum_{1\le j\le k} s_{ij}\mathbf{b}_j^*$. Choose i such that $r_{ik} \neq 0$. Notice that the r_{ij} are integers and that $r_{ik} = s_{ik}$. Since $\mathbf{x}_1, \ldots, \mathbf{x}_t$ are independent we have $k \ge t$. Observe that

$$|\mathbf{x}_i|^2 \ge s_{ik}^2|\mathbf{b}_k^*|^2 = r_{ik}^2|\mathbf{b}_k^*|^2 \ge |\mathbf{b}_k^*|^2.$$

So, whenever $j < k$,

$$|\mathbf{b}_j|^2 \le 2^{k-1}|\mathbf{b}_k^*|^2 \le 2^{k-1}|\mathbf{x}_i|^2 \le 2^{r-1}\max(|\mathbf{x}_1|^2, \ldots, |\mathbf{x}_t|^2).$$

In particular, since $k \ge t$, this inequality holds whenever $j \le t$.

Let us now give an informal description of the LLL-reduction procedure applied to any k-tuple of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$.

Algorithm 4.4 (LLL-Reduction). Suppose that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}$ are LLL-reduced (true if $k = 2$). Replace $\mathbf{b}_k$ by $\mathbf{b}_k - \sum_{j<k} a_j\mathbf{b}_j$ with $a_j \in \mathbb{Z}$ in such a way that $|\mu_{k,j}| \le 1/2$ whenever $j < k$. Suppose $|\mathbf{b}_k^*|^2 \ge (3/4 - \mu_{k,k-1}^2)|\mathbf{b}_{k-1}^*|^2$. Then $\mathbf{b}_1, \ldots, \mathbf{b}_k$ is LLL-reduced and we can stop. If the Lovász condition is not satisfied we interchange $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$, apply LLL-reduction to $\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}$ and repeat the procedure.

Now apply LLL-reduction to the basis $\mathbf{b}_1, \ldots, \mathbf{b}_r$ of a lattice L. It is clear that if the algorithm terminates we have obtained an LLL-reduced basis of L. It remains to show that the algorithm actually terminates. To this end we introduce the quantities

$$d_i = \det((\mathbf{b}_s \cdot \mathbf{b}_t)_{s,t=1,\ldots i})$$

for $i = 1, \ldots, r$. In particular, $d_r = d(L)^2$. Let

$$D = \prod_{i=1}^{r-1} d_i.$$

During the LLL-reduction this quantity changes only value when two vectors $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ are interchanged. In fact, only d_{k-1} changes value in that case. A simple computation shows that the new value will be $d'_{k-1} = d_{k-1}|\mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*|^2/|\mathbf{b}_{k-1}^*|^2$. Since we had to interchange $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ the Lovász condition is apparently not satisfied and so we get $d'_{k-1} \le \frac{3}{4}d_{k-1}$. Hence D gets reduced by a factor 3/4. Note that D has a lower bound which depends only on the lattice and not on the choice of basis. This can be seen as follows. Let l be the length of the shortest nonzero vector in L. Minkowski's theorem applied to the lattice generated by the first i vectors shows that $l \leq \sqrt{i} d_i^{1/2i}$ for each $i = 1, \dots, r$. Hence $d_i \geq (l^2/i)^i$ and so, $D \geq (l^2/r)^{r(r-1)/2}$. In particular, we see that the number of interchanges in the LLL-reduction is bounded by $O(\log D + r^2 \log(\sqrt{r}/l))$ and so the algorithm terminates.

Exercise 4.5. Let $m = \max_{i=1,\dots,r} |\mathbf{b}_i|$. Show that the number of swaps occurring in the LLL-algorithm is bounded by $cr^2 \log(m\sqrt{r}/l)$ where $c > 0$ is a constant and l is the length of the shortest nonzero vector in L.

5. Implementations of the LLL-Algorithm

It turns out to be possible to give very simple implementations of the LLL-algorithm. Here we shall give a version which requires only operations on the Gram-matrix and an auxiliary matrix which keeps track of the transformation between the original basis and the transformed basis. Our first observation is that the matrix $(\mathbf{b}_i \cdot \mathbf{b}_j^*)_{i,j=1,\dots,r}$ can be obtained by putting the Gram-matrix of $(\mathbf{b}_i \cdot \mathbf{b}_j)$ into column echelon form by the algorithm **Gram-Schmidt** sketched above. The second observation is that replacement of $\mathbf{b}_k$, say, by $\mathbf{b}_k - \sum_{j<k} a_j \mathbf{b}_j$ does not change the corresponding vectors $\mathbf{b}_i^*$. The third observation is more subtle. If we interchange $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$ and apply Gram-Schmidt to the newly ordered set, we obtain a new orthogonal system $\{\mathbf{b}_i^{**}\}_i$. Notice however, that $\mathbf{b}_i^{**} = \mathbf{b}_i^*$ if $i \neq k, k-1$ and that

$$\mathbf{b}_{k-1}^{**} = \mathbf{b}_k^* + \frac{\mathbf{b}_k \cdot \mathbf{b}_{k-1}^*}{|\mathbf{b}_{k-1}^*|^2} \mathbf{b}_{k-1}^*$$

and

$$\begin{aligned} \mathbf{b}_k^{**} &= \mathbf{b}_{k-1}^* - \frac{\mathbf{b}_{k-1} \cdot \mathbf{b}_{k-1}^{**}}{|\mathbf{b}_{k-1}^{**}|^2} \mathbf{b}_{k-1}^{**} \\ &= \mathbf{b}_{k-1}^* - \frac{\mathbf{b}_k \cdot \mathbf{b}_{k-1}^*}{|\mathbf{b}_{k-1}^{**}|^2} \mathbf{b}_{k-1}^{**}. \end{aligned}$$

Based on these observations we can propose the following implementation. Suppose that we want to carry out LLL-reduction on the vectors $\mathbf{b}_1, \dots, \mathbf{b}_n$. We introduce the $n \times n$-matrix H to keep a record of the relation between the (partially) reduced set of vectors and the original $\mathbf{b}_i$. We initialise H either to the matrix whose rows are the $\mathbf{b}_i$ or to the $n \times n$-identity matrix. The $n \times n$-matrix G will be used to carry out the reduction. We initialise it by the reduced Gram matrix $(\mathbf{b}_i \cdot \mathbf{b}_j^*)$. For the purpose of the algorithm we concatenate the matrices G and H to $(G|H)$. We have two procedures which will be the building blocks of our LLL-reduction. These procedures affect the matrix $(G|H)$ which is assumed to be a global variable. The integer k in the input is assumed to satisfy $1 \leq k \leq n$.

Procedure **reduce**(k,l). We assume that $l < k$ and that the G-part of $(G|H)$ is in lower triangular form. If $G_{l,l} = 0$ we do nothing. If $G_{l,l} \neq 0$ we choose the nearest integer q to $G_{k,l}/G_{l,l}$ and subtract q times the l-th row of $(G|H)$ from the k-th row of $(G|H)$.
Procedure **swap**(k). We assume that the G-part of $(G|H)$ is in lower diagonal form. Interchange the k-th and $k-1$-st row in $(G|H)$. Interchange the k-th and $k-1$-st column in G. Add $G_{k-1,k}/G_{k,k}$ times the k-th column to the $k-1$-st column of G. Add a suitable multiple of the $k-1$-st column of G to the k-th column so that the element at place $k-1, k$ becomes zero.

The LLL-algorithm proceeds as follows. We initialise G and H to the reduced Gram matrix and the matrix of $\mathbf{b}_i$'s respectively and then apply the following procedure with $k = n$.

Algorithm 5.1 (LLL).

```
Procedure LLL($k$):
  if $k = 1$ then stop fi;
  LLL($k-1$); lovasz := false;
  while lovasz = false do
    if $G_{k-1,k-1} = 0$ then stop fi;
    reduce($k, k-1$);
    $\mu := G_{k-1,k}/G_{k-1,k-1}$;
    lovasz := $(G_{k,k} \geq (\frac{3}{4} - \mu^2)G_{k-1,k-1})$;
    if lovasz = true
    then for $l$ from 1 to $k-1$ do reduce($k, k-l$); od;
    else swap($k$); LLL($k-1$);
    fi;
  od.
```

If we do not know the $\mathbf{b}_i$ explicitly, but only the Gram-matrix, we can initialise H to the $n \times n$ identity matrix. After finishing the algorithm the matrix H will be the transformation matrix between the $\mathbf{b}_i$ and the reduced basis.

One may notice that we can apply this algorithm without any change to a set of vectors $\mathbf{b}_i$ which is not necessarily $\mathbb{R}$-linearly independent, but where it is known that they generate a (discrete) lattice. Let $\mathbf{b}'_1, \ldots, \mathbf{b}'_n$ be the outcome of **LLL**. If the rank of the $\mathbf{b}_i$ is r, then $\mathbf{b}'_i = 0$ for $i = 1, \ldots, n-r$ and the remaining $\mathbf{b}'_i$ will be a reduced basis of the lattice generated by the $\mathbf{b}_i$. In the case of dependent input vectors we have to recheck our termination proof of **LLL**. However, we can simply use the quantities $d'_k = \prod_{i=1, G_{i,i} \neq 0}^{k} G_{i,i}$ instead of the d_k.

6. Small Linear Forms

The first application was by its inventors, who used it to construct a polynomial time algorithm to factor polynomials. In the section on factorisation of polynomials we shall discuss it. The application we have in mind here is finding $\mathbb{Z}$-linear combinations of a given set of real numbers with very small values. What is meant by 'small' is indicated by the following theorem of Dirichlet.

Theorem 6.1. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and $A = \sum_{i=1}^{r} |\alpha_i|$. For every $M \in \mathbb{N}$ there exist $m_1, \ldots, m_n \in \mathbb{Z}$, not all zero, such that $|m_i| < M$ for all i and $|m_1\alpha_1 + \cdots + m_n\alpha_n| < A/M^{n-1}$.*

Proof. We may assume that $\alpha_i \geq 0$ for all i. Consider the set .

$$B = \{k_1\alpha_1 + \cdots + k_n\alpha_n \mid \forall i : \; k_i \in \mathbb{Z}, \; 0 \leq k_i < M\}.$$

Note that $\#B = M^n$ and $0 \leq x \leq A(M-1)$ for each $x \in B$. Divide the interval $[0, A(M-1)]$ into $M^n - 1$ subintervals of equal length. Since $\#B = M^n$ there exists at least one interval containing at least two elements of B, say $k_1\alpha_1 + \cdots + k_n\alpha_n$ and $k'_1\alpha_1 + \cdots + k'_n\alpha_n$. We have applied the so-called 'box principle' or 'pigeon hole principle' here. Let $m_i = k_i - k'_i$ for all i; then we conclude that $|m_1\alpha_1 + \cdots + m_n\alpha_n| \leq A(M-1)/(M^n-1) < A/M^{n-1}$ and not all m_i are zero.

Dirichlet's theorem is optimal in the following sense.

Theorem 6.2. *Let $\epsilon > 0$ and let V_ϵ be the subset all real n-tuples $(\alpha_1, \ldots, \alpha_n)$ with $\sum_{i=1}^{r} |\alpha_i| = 1$ such that the inequality*

$$|m_1\alpha_1 + \cdots + m_n\alpha_n| < 1/M^{n-1+\epsilon}$$

hase infinitely many solutions $m_1, \ldots, m_n \in \mathbb{Z}$ with $|m_i| < M$ for all i. Then V_ϵ has measure zero.

Let $\alpha_1, \ldots, \alpha_n$ be real numbers normalised such that, say, $\sum_{i=1}^{n} |\alpha_i|^2 = 1$. A very important application of the LLL-method is to prove the non-existence of integers $m_1, \ldots, m_n$ such that

$$|m_1\alpha_1 + \cdots + m_n\alpha_n| < \epsilon, \quad |m_1|, \ldots, |m_n| < M \tag{6.1}$$

where the product ϵM^{n-1} is extremely small. So we are in an exceptional case with respect to Dirichlet's theorem.

To this end choose $N = M/\epsilon$ and apply LLL-reduction to the lattice L generated by the row vectors of the matrix

$$C = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & N\alpha_1 \\ 0 & 1 & 0 & \cdots & 0 & N\alpha_2 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & N\alpha_n \end{pmatrix}.$$

With the standard inner product on $\mathbb{R}^{n+1}$, the determinant of the lattice is given by $d(L)^2 = 1 + N^2(\alpha_1^2 + \cdots + \alpha_n^2) = 1 + N^2$. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be the reduced basis we have found. Suppose that a solution to (6.1) exists. Denote $(m_1, \ldots, m_n, m_1 N\alpha_1 + \cdots + m_n N\alpha_n)$ by $\mathbf{x}$ and note that $|\mathbf{x}| \leq (n+1)M$.

From the properties of a reduced basis it would then follow that $|\mathbf{b}_1| \leq 2^{(n-1)/4}(n+1)M$. So, if this inequality does not hold, we have a contradiction and conclude that (6.1) does not have any solutions. Note that $M = \epsilon N \leq \epsilon d(L)$, hence $M^n \leq (\epsilon M^{n-1})d(L)$. Since ϵM^{n-1} is exceptionally small, we see that $M/d(L)^{1/n}$, and hence $|\mathbf{b}_1|/d(L)^{1/n}$ are exceptionally small. Since this does not happen in generic cases we can usually disprove the existence of solutions to (6.1).

In the explicit solution of diophantine equations using Gel'fond-Baker techniques we typically find the problem

$$|m_1\alpha_1 + \cdots + m_n\alpha_n| < c^{-\max_i |m_i|}, \quad |m_i| < M$$

for some $c > 1$. Because of the inequalities $|m_i| < M$ we have a finite search range for solutions. However, due to the size of M a case by case check is impossible. Instead we will use a technique developed by B.M.M. de Weger [4] in his Ph.D. thesis.

Let us first look for solutions with $\max_i |m_i| > 2n\log(M)/\log(c)$. For such solutions we get the inequalities (6.1) with $\epsilon = M^{-2n}$. We now apply the LLL-algorithm in the way described above. If M is large, ϵ is extremely small compared to the Dirichlet bound. Hence we expect our method to show that there are no solutions of (6.1). This expectation is practically always vindicated. So the only solutions of our original problem satisfy

$$|m_1\alpha_1 + \cdots + m_n\alpha_n| < c^{-\max_i |m_i|}$$

with $|m_i| \leq 2n\log(M)/\log(c)$. Note that for large M this is a spectacular reduction of our search space. If so desired we repeat the procedure with $2n\log(M)/\log(c)$ to get an even further reduction of the search space. In solving diophantine equations one has usually two or three of these rounds. This technique has been applied in many papers on computational diophantine equations. As an illustration we mention the paper [3].

Notes

This chapter was prepared as material for a course at the Master Class Computational Number Theory of the Mathematical Research Institute in the Netherlands in 1995. During the preparation we made good use of H. Cohen's beautiful book [1] and profited from many discussions with Wilberd van der Kallen.

References

1. H. Cohen (1995): *A Course in Computational Algebraic Number Theory* (2nd edition) Springer-Verlag, Berlin Heidelberg New York.
2. A. K. Lenstra, H. W. Lenstra jr., and L. Lovász (1982): *Factoring polynomials with rational coefficients*, Math. Ann. **261**, 515–534.
3. N. Tzanakis and B. M. M. de Weger (1989): *On the practical solution of the Thue equation*, J. Number Theory **31**, 99–132.
4. B. M. M. de Weger (1987): *Solving exponential diophantine equations using lattice basis reduction algorithms*, J. Number Theory **26**, 325–367.

Chapter 4. Factorisation of Polynomials

Frits Beukers

1. Introduction

In many rings, commutative or not commutative, the elements can be written as a product of irreducible elements (not necessarily unique). In algorithms in computer algebra it is often essential that this should be realized in an efficient way. The most important examples in this respect are $\mathbb{Z}$, $\mathbb{F}_q[X]$, $\mathbb{Z}[X]$ and, as a more recent example of computational interest, the ring $\mathbb{Q}(X)[d/dX]$ [6]. The latter ring is not commutative and factorisation into irreducibles is not unique.

Factorisation in $\mathbb{Z}$, or rather our apparent inability to do this efficiently, lies at the heart of new developments in cryptography and random number generation. Factorisation in $\mathbb{Q}(X)[d/dX]$ has gained some interest recently because it enables one to find algebraic relations between solutions of linear differential equations [2]. In this lecture we shall concentrate on factorisation in $\mathbb{F}_q[X]$ and $\mathbb{Z}[X]$.

2. Berlekamp's Algorithm

Let $\mathbb{F}_q$ be the finite field with $q = p^a$ elements for some prime p and let $f \in \mathbb{F}_q[X]$. The problem is to find distinct irreducible polynomials $f_1, \ldots, f_r \in \mathbb{F}_q[X]$ and $e_i \in \mathbb{N}$ such that $f = f_1^{e_1} \cdots f_r^{e_r}$. We call the powers $f_i^{e_i}$ the *primary factors* of f. When q is small an efficient and widely used algorithm for decomposition into primary factors is Berlekamp's algorithm. It is based on the following observation

Lemma 2.1. *Let $v \in \mathbb{F}_q[X]$ be such that $v^q \equiv v \bmod f$. Then*

$$f = \prod_{a \in \mathbb{F}_q} \gcd(f, v - a).$$

Proof. It is known that $Y^q - Y = \prod_{a \in \mathbb{F}_q} (Y - a)$. Hence

$$f = \gcd(f, v^q - v) = \gcd\left(f, \prod_{a \in \mathbb{F}_q} (v - a)\right) = \prod_{a \in \mathbb{F}_q} \gcd(f, v - a).$$

The latter equality follows from the fact that $\gcd(v - a, v - b) = 1$ whenever $a \neq b$.

For the computation of a gcd we can use Euclid's algorithm which works very efficiently in $\mathbb{F}_q[X]$. Of course, a factor of the form $\gcd(f, v-a)$ need not be irreducible or even primary. But of course we can repeat the procedure to each of these factors with a different solution of $v^q \equiv v \bmod f$. The second observation which makes Berlekamp's algorithm work is that if we use sufficiently many distinct v we obtain a factorisation of f into primary factors. This will be proved below. The third observation is that solving $v^q \equiv v \bmod f$ is basically an $\mathbb{F}_q$-linear problem since $v(X)^q = v(X^q)$. In fact, the space of solutions to $v^q \equiv v \bmod f$ is an $\mathbb{F}_q$-linear vector space.

To determine the dimension of this vector space we use the ring isomorphism

$$\mathbb{F}_q[X]/(f) \cong \oplus_{i=1}^r \mathbb{F}_q[X]/(f_i^{e_i}) \qquad (C)$$

given by the Chinese Remainder Theorem. We have

Theorem 2.2. *The set of solutions v mod f to $v^q \equiv v$ mod f forms a ring and is isomorphic, via (C), to the subring $\oplus_{i=1}^r \mathbb{F}_q$ of $\oplus_{i=1}^r \mathbb{F}_q[X]/(f_i^{e_i})$. In particular, it is an $\mathbb{F}_q$-linear vector space of dimension r.*

Proof. We determine the solution set of $v^q = v$ in $\oplus_{i=1}^r \mathbb{F}_q[X]/(f_i^{e_i})$. It suffices to prove that, for any $v \in \mathbb{F}_q[X]$ and any $i = 1, \ldots, r$, we have

$$v^q \equiv v \bmod f_i^{e_i} \iff \exists s \in \mathbb{F}_q :\ v \equiv s \bmod f_i^{e_i}.$$

The proof of '$\Leftarrow$' being trivial, we note that $f_i^{e_i} = \prod_{a \in \mathbb{F}_q} \gcd(f_i^{e_i}, v-a)$. Since the factors in the product are pairwise relatively prime and since f_i is irreducible there is at most one a for which $\gcd(f_i^{e_i}, v-a)$ is nontrivial. Hence $f_i^{e_i}$ divides $v - a$ and we are done.

Exercise 2.3. Prove that the solutions in $\mathbb{F}_q[X]/(f)$ of $v^p = v$ form an r-dimensional vector space over $\mathbb{F}_p$.

In an informal but unambiguous way we can now describe Berlekamp's algorithm as follows. Determine a basis $v_1 = 1, v_2, \ldots, v_r$ of $v^q \equiv v \bmod f$. Let $E = \{f\}$ (E will be a set of factors of f whose product equals f). If $r = 1$ we are done, f is primary. If $r > 1$, we replace for $j = 2, \ldots, r$ each element $h \in E$ by the nontrivial elements of the set $\{\gcd(h, v_j - a)\}_{a \in \mathbb{F}_q}$. Obviously, the algorithm terminates. The resulting set E is the set of primary factors of f. To see the latter statement consider an element $h \in E$. For every j there exists $s_j \in \mathbb{F}_q$ such that $v_j \equiv s_j \bmod h$. A fortiori, since $v_1, \ldots, v_r$ is a basis, to every solution v of $v^q \equiv v \bmod f$ there exists s_v such that $v \equiv s_v \bmod h$. Now suppose that h contains two relatively prime primary factors, say $f_1^{e_1}$ and $f_2^{e_2}$. Then, as v runs through all solutions of $v^q = v$, we have $v \equiv s_v \bmod f_i^{e_i}$ for $i = 1, 2$. In particular, the solutions to $v^q = v$ do not surject to $\oplus_{i=1}^r \mathbb{F}_q[X]/(f_i^{e_i})$, which is a contradiction.

Exercise 2.4. Formulate a variant of Berlekamp's algorithm where we use the solutions of $v^p = v$. What are the possible (dis)advantages?

A problem which remains after performing Berlekamp's algorithm is to decompose the primary factors into irreducible factors. This is easy. If the primary factor, say F, is not a polynomial in X^p, we recover the irreducible factor by F/F' where F' is the derivative of F. If F is a polynomial in X^p first write $F = g^{p^k}$ where g is not a polynomial in X^p. Then determine g/g'.

An alternative, which seems more economic, is to reduce f to a squarefree polynomial first and then apply Berlekamp's algorithm. Suppose that $f = gh^p$ where g is a p-th power free polynomial. Then the squarefree polynomial $\tilde{f}$ defined by $\tilde{f} = f/\gcd(f, f')$ has the same irreducible factors as g. Here f' denotes the derivative of f. To determine these irreducible factors we can feed $\tilde{f}$ to Berlekamp's algorithm. Having found the irreducible factors of g we can divide them out from f and we are left with the pure power h^p in which we have to factor h via the same method.

Finally we add a word on the implementation of the solution of $v^q \equiv v \bmod f$. Let n be the degree of f. Consider the basis $1, X, X^2, \ldots, X^{n-1}$ of the $\mathbb{F}_q$-vector space $\mathbb{F}_q[X]/(f)$. With respect to this basis the linear map sending any v to its q-th power v^q has a matrix which we denote by Q. We formalise this in a lemma.

Lemma 2.5. *Let $f \in \mathbb{F}_q[X]$ be a polynomial of degree n. Let Q be the $n \times n$-matrix whose i-th row contains the coefficients of the polynomial X^{iq} reduced modulo f. Then, for any polynomial v of degree $< n$ and coefficient vector $\mathbf{v}$, the coefficient vector of v^q reduced modulo f equals $\mathbf{v} \cdot Q$.*

The computation of $v_1, \ldots, v_r$ then comes down to computation of a basis of the kernel of Q.

Exercise 2.6. Let p be the characteristic of $\mathbb{F}_q$ and let $f \in \mathbb{F}_q[X]$. Prove that f is the p-th power of another polynomial if and only if $f \in \mathbb{F}_q[X^p]$.

Exercise 2.7. Let $f, g, h \in \mathbb{F}_q[X]$ and $f = gh^p$ where g does not contain any p-th power of a polynomial. Prove that we can find h^p by the following algorithm. Repeat $f := \gcd(f, f')$ until $f' = 0$. Then $f = h^p$. What is the maximal number of steps required?

Exercise 2.8. Let $f_1, \ldots, f_r$ be the irreducible factors of f whose exponent in the factorisation of f is not disivible by p. Prove that $f_1 \cdots f_r = f/\gcd(f, f')$.

Exercise 2.9. Factor $X^{12} - 1$ in $\mathbb{F}_5[X]$ by hand using Berlekamp's algorithm.

Exercise 2.10. Here is another method to produce solutions of the congruence $v^q \equiv v \bmod f$. Assume that f is squarefree and has irreducible factors all of the same degree s. Prove that, for an arbitrary polynomial b, the polynomial

$$b^{1+q+q^2+\cdots+q^{s-1}} \bmod f$$

is such a solution. (Hint: use Lemma 3.1 of the next section).

3. Additional Algorithms

It is clear that the bottleneck of Berlekamp's algorithm lies in the determination of the factorisation $h = \prod_{a\in\mathbb{F}_q} \gcd(h, v-a)$. For every $a \in \mathbb{F}_q$ we have to determine a gcd. In particular, when q is large compared to r, most gcd-computations of $\gcd(h, v-a)$ will yield a trivial result, which is of course quite wasteful. A variant of Berlekamp's algorithm which does not have this defect is the Cantor-Zassenhaus algorithm. However, it is of a probabilistic nature, but this need not disturb a practically minded factoriser. Let notation be as in the previous section, suppose that q is odd and that $r > 1$. Suppose we have randomly chosen a solution $v \in \mathbb{F}_q$ of $v^q \equiv v \bmod f$, whatever 'random' means. Consider the factorisation $f = \gcd(f, v)\gcd(f, v^{(q-1)/2}-1)\gcd(f, v^{(q-1)/2}+1)$. This factorisation is only trivial if f divides either of the three factors, and since we may reasonably assume $f \not| v$ we have a trivial factorisation if f divides either $v^{(q-1)/2}-1$ or $v^{(q-1)/2}+1$. Define $s_i \in \mathbb{F}_q$ by $v \equiv s_i \bmod f_i^{e_i}$ for $i = 1, \ldots, r$. Then we have a trivial factorisation if and only if all s_i are simultaneously squares, or simultaneously non-squares. The probability that this happens is of course

$$2\left(\frac{\frac{q-1}{2}}{q}\right)^r < \frac{1}{2^{r-1}} \leq \frac{1}{2}.$$

Hence the probability of failure is at most $1/2$ for one trial, and at most $1/1000$ for a row of ten trials! For many practical purposes this idea suffices to find the complete factorisation of f.

We mention one more way of obtaining (partial) factorisations of f. It is based on the following well-known lemma.

Lemma 3.1. *The polynomial $X^{q^k} - X$ factors over $\mathbb{F}_q[X]$ into the product of all monic irreducible polynomials of degree dividing k.*

For squarefree f of degree n this lemma yields the following factorisation algorithm. Let $g_1 = \gcd(X^q - X, f)$ and define recursively

$$g_i = \gcd\left(X^{q^i} - X, f/\prod_{j=1}^{i-1} g_j\right), \qquad i = 2, \ldots, n.$$

Then we have $f = g_1 \cdots g_n$, where g_i is the product of all irreducible factors of f of degree i. We call this the ***distinct degree factorisation.*** In practice we can apply this method by computing $X^q \bmod F$ and $X^{q^i} \equiv (X^{q^{i-1}})^q \bmod f$

for $i = 2, \ldots, [n/2]$. This repeated procedure of taking q-th powers can be done quite efficiently using the matrix Q introduced in Lemma 2.5.

As a final remark we note that the discussions above also provide us with two irreducibility tests for elements of $\mathbb{F}_q[X]$. We collect them in the following proposition together with a third one.

Proposition 3.2. *Suppose that $f \in \mathbb{F}_q[X]$ is squarefree of degree n. Let Q be the matrix constructed in Lemma 2.5. Then f is irreducible if and only if at least one of the following conditions hold.*

1. *The rank of the kernel of $Q - \mathrm{Id}$ equals 1.*
2. $\gcd(f, X^{q^i} - X) = 1$ *for* $i = 1, \ldots, [n/2]$.
3. $f | (X^{q^n} - X)$ *and* $\gcd(X^{q^{n/l}} - X, f) = 1$ *for every prime l dividing n.*

Exercise 3.3. Prove the validity of the irreducibility tests in the above proposition.

Most computer algebra packages have a built-in algorithm for factorisation of polynomials f in $\mathbb{F}_p[X]$, where p is a prime. We have

`Factor[f,Modulus->p]`	in	*Mathematica*,
`factmod(f,p)`	in	PARI and
`Factor(f) mod p`	in	Maple.

Exercise 3.4. Play with one or more of the above functions and try to find how large the degree of f or the size of p can be taken before running times tend to become long.

4. Polynomials with Integer Coefficients

We start by noting that if $P \in \mathbb{Z}[X]$ factors in $\mathbb{Q}[X]$, then this factorisation takes place in $\mathbb{Z}[X]$. This fact follows from the so-called *Lemma of Gauss*. By the *content* of P we simply mean the gcd of the coefficients of P. We denote it by $c(P)$. A polynomial with content 1 is called *primitive*.

Lemma 4.1 (Gauss). *Let $A, B, C \in \mathbb{Z}[X]$ and $C = AB$. Then*

$$c(C) = c(A)c(B).$$

Proof. We may as well assume that $c(A) = c(B) = 1$. We have to show that $c(C) = 1$. Suppose $c(C) > 1$ and let p be a prime divisor of $c(C)$. Then $C \equiv 0 \bmod p$ hence $AB \equiv 0 \bmod p$. But this implies that $\mathbb{F}_p[X]$ has zero divisors, which is impossible. Hence $c(C) = 1$.

Corollary 4.2. *Let $P \in \mathbb{Z}[X]$ and suppose $P = AB$ with $A, B \in \mathbb{Q}[X]$. Then there exist $a, b \in \mathbb{Q}$ such that $aA, bB \in \mathbb{Z}[X]$ and $P = (aA)(bB)$.*

Proof. Without loss of generality we may assume $c(P) = 1$. Choose $a, b \in \mathbb{Q}$ such that aA, bB are polynomials with integral coefficients and content 1. From Gauss's Lemma we obtain $c(abP) = c(aB)c(bB) = 1$. So abP is a polynomial with integral coefficients and content 1. Since $c(P) = 1$ we conclude that $ab = \pm 1$. Choosing the sign of a suitably we can see to it that $ab = 1$ and hence $P = (aA)(bB)$.

So when factoring in $\mathbb{Q}[X]$ we might as well restrict to factorisation problems in $\mathbb{Z}[X]$. We know that a divisor d of an integer a has the property that $|d| \le |a|$. In $\mathbb{Z}[X]$ we have a similar property with respect to the l_2-norm. For any $f = \sum_{i=0}^n f_i X^i \in \mathbb{Z}[X]$ we denote $||f|| = (\sum_{i=0}^n f_i^2)^{1/2}$.

Theorem 4.3 (Landau-Mignotte). *Let $f, g \in \mathbb{Z}[X]$ have degrees n and m respectively. Suppose that g divides f. Then, for $i = 0, \ldots, m$, we have $|g_i| \le \binom{m}{i} ||f||$. Moreover,*

$$||g|| \le \binom{2m}{m}^{1/2} ||f||.$$

Proof. First of all we establish for any $a \in \mathbb{C}$ and $h \in \mathbb{C}[X]$,

$$||(X - a)h|| = |a| \cdot ||(X - \bar{a}^{-1})h||.$$

This can be seen as follows. Put $h = \sum_{i=0}^n h_i X^i$ and $h_{-1} = h_{n+1} = 0$. Then,

$$\begin{aligned} ||(X-a)h||^2 &= \sum_{i=0}^{n+1} |h_{i-1} - ah_i|^2 \\ &= \sum_{i=0}^{n+1} (|h_{i-1}|^2 - ah_i\bar{h}_{i-1} - \bar{a}h_{i-1}\bar{h}_i + |ah_i|^2) \\ &= \sum_{i=0}^{n+1} (|h_i|^2 - ah_i\bar{h}_{i-1} - \bar{a}h_{i-1}\bar{h}_i + |ah_{i-1}|^2) \\ &= \sum_{i=0}^{n+1} |\bar{a}h_{i-1} - h_i|^2 \\ &= ||(\bar{a}X - 1)h||^2 = |a|^2 ||(X - \bar{a}^{-1})h||^2. \end{aligned}$$

Let $b_1, \ldots, b_s$ be the set of zeros of f outside of the unit disk in the complex plane ordered in decreasing absolute value. Let $a_1, \ldots, a_{n-s}$ be the set of zeros inside the unit disk. Then,

$$\begin{aligned} ||f||^2 &= |a_1 \cdots a_{n-s}|^2 ||f_n \prod_{i=1}^{n-s} (X - \bar{a}_i^{-1}) \prod_{j=1}^{s} (X - b_j)||^2 \\ &\ge |f_n b_1 \cdots b_s|^2. \end{aligned}$$

Hence $||f|| \geq |f_n b_1 \cdots b_s|$. Let $\gamma_1, \ldots, \gamma_m$ be the set of zeros of g ordered in decreasing absolute value. The i-th coefficient of g equals

$$g_i = (-1)^i g_m \sigma_i(\gamma_1, \ldots, \gamma_m),$$

where σ_i is the i-th symmetric function. Since this function contains $\binom{m}{i}$ products of γ_j we obtain

$$\begin{aligned} |\sigma_i(\gamma_1, \ldots, \gamma_m)| &\leq \binom{m}{i} |\gamma_1 \cdots \gamma_i| \\ &\leq \binom{m}{i} |b_1 \cdots b_i| \leq \binom{m}{i} |b_1 \cdots b_s| \\ &\leq \binom{m}{i} ||f||/|f_n|. \end{aligned}$$

Furthermore, g_m divides f_n, whence $|g_m| \leq |f_n|$, and the result follows. The inequality $||g|| \leq \binom{2m}{m}^{1/2} ||f||$ follows from the identity $\sum_{i=0}^m \binom{m}{i}^2 = \binom{2m}{m}$.

One might wonder if the bound $\binom{2m}{m}$ is not too large. An interesting theorem by Mignotte [5] shows that in fact it is not.

Theorem 4.4 (Mignotte). *There exists $c > 0$ with the following property. For every $m \in \mathbb{N}$ there exist $f, g \in \mathbb{Z}[X]$ with $g|f$ and $\deg(g) = m$ such that*

$$||g|| > \binom{2m}{m}^{1/2} \frac{c||f||}{m\sqrt{\log(m)}}.$$

The Landau-Mignotte bound reduces the set of polynomials which may possibly divide a polynomial $f \in \mathbb{Z}[X]$ to a finite set. This gives an effective, but highly inefficient, factorisation algorithm. However, the bound can be used in two efficient factorisation algorithms, which we present in the following sections.

Exercise 4.5. An ad hoc way to prove irreducibility of a polynomial in $\mathbb{Z}[X]$ is to prove that its reduction modulo a prime p is irreducible in $\mathbb{F}_p[X]$. For arbitrary irreducible polynomials such a prime can usually be found very quickly. However, there are exceptions. Prove that $X^4 + 1$ is irreducible in $\mathbb{Z}[X]$ but reducible modulo any prime.

5. Factorisation of Polynomials with Integer Coefficients, I

The first, and oldest, algorithm we present starts with a factorisation of the polynomial f reduced modulo some prime p, which is then (almost) lifted to characteristic zero via the so-called *Hensel lift*.

Theorem 5.1 (Hensel's Lemma). *Let $f, g, h \in \mathbb{Z}[X]$ be monic polynomials. Let p be a prime and $k \in \mathbb{N}$ such that $f \equiv gh \,(\mathrm{mod}\, p^k)$ and* $\gcd(g \bmod p,\ h \bmod p) = 1$. *Then there exist monic polynomials $\tilde{g}, \tilde{h}$ such that $\tilde{g} \equiv g \,(\mathrm{mod}\, p^k)$, $\tilde{h} \equiv h \,(\mathrm{mod}\, p^k)$ and $f \equiv \tilde{g}\tilde{h} \bmod p^{k+1}$. Moreover, $\tilde{g}$ and $\tilde{h}$ are uniquely determined modulo p^{k+1}.*

Proof. Put $\tilde{g} = g + p^k u$ and $\tilde{h} = h + p^k v$, where u, v are polynomials to be determined of degrees less than $\deg(g), \deg(h)$ respectively. The equation $f \equiv \tilde{g}\tilde{h} \bmod p^{k+1}$ comes down to $f - (g + p^k u)(h + p^k v) \equiv 0 \pmod{p^{k+1}}$, hence

$$\frac{f - gh}{p^k} - (uh + gv) \equiv 0 \pmod{p}.$$

Since $g \bmod p$ and $h \bmod p$ are relatively prime, there are (mod p) uniquely determined u, v of degrees less than $\deg(g)$ and $\deg(h)$ respectively such that $uh + gv \equiv (f - gh)/p^k \bmod p$. A fortiori $\tilde{g}$ and $\tilde{h}$ are uniquely determined modulo p^{k+1}.

The upshot of this theorem is that a mod p factorisation into relatively prime factors can be lifted to a factorisation in the so-called p-adic numbers. Notice that the proof given above also provides a simple algorithm to perform such a lift. In fact, there is very simple variation on this lifting which can be described as follows.

```
$Hensel(f, g, h)$ =
    **choose** $a, b \in \mathbb{Z}[X]$ such that $ag + bh \equiv 1 \pmod{p}$;
    $g_1 := g$;
    **for** $k = 1, 2, 3, 4, \ldots$ **do**
        Determine $u_k \in \mathbb{Z}[X]$ such that
        $\deg(u_k) < deg(g)$ and $u_k \equiv bf \pmod{g_k}$;
        $g_{k+1} := g_k + u_k$;
    **od**.
```

Exercise 5.2. Show that, in the above algorithm, we have

$$f \equiv 0 \pmod{p^k, g_k}$$

for every k.

There is a simple way to speed up the Hensel lifting procedure and go from a mod p^k factorisation to a mod p^{2k} factorisation in one step.

Exercise 5.3. Modify the proof of Hensel's lemma to go from mod p^k to mod p^{2k} in one step.

We can now give a factorisation algorithm for $f \in \mathbb{Z}[X]$. Let us assume that $c(f) = 1$ and that f is squarefree. Let n be the degree of f and let B be an a priori upper bound for the absolute values of the coefficients of possible divisors of f whose degree is at most $[n/2]$. This can either be given by the Landau-Mignotte bound or some other estimate. Let f_n be the leading coefficient of f. Then perform the following steps.

1. Choose a prime p not dividing the discriminant of f and f_n. Factor f modulo p.
2. Choose k so that $p^k > 2|f_n|B$ and lift the mod p factorisation to a factorisation $f \equiv f_n \prod_{i=1}^{r} h_i \bmod p^k$ where the h_i are monic.
3. For each subset $S \subset \{1,\dots,r\}$ compute $h \in \mathbb{Z}[X]$ such that $h \equiv f_n \cdot \prod_{i\in S} h_i \bmod p^k$ and such that h has degree at most $[n/2]$, coefficients less than $|f_n|B$ in absolute value and test whether h divides $f_n \cdot f$.

Although the above algorithm works satisfactory in most cases, it is not guaranteed to have a running time polynomial in n. Consider for example the extreme case where f factors into linear factors modulo p. In the final step we then have to make roughly 2^n verifications, which is indeed exponential in n. One might argue that if a factorisation modulo p has too many small factors one could take another prime p. However, even this is not always a solution as shown by the following construction. Let $d_1,\dots,d_r$ be a set of pairwise relatively prime integers and define

$$f_{d_1,\dots,d_r}(X) = \prod_{\epsilon_1,\dots,\epsilon_r\in\pm1} (X - \epsilon_1\sqrt{d_1} - \cdots - \epsilon_r\sqrt{d_r}) \in \mathbb{Z}[X].$$

Exercise 5.4. Prove that modulo any prime p, not dividing $d_1\cdots d_r$, the polynomial $f_{d_1,\dots,d_r}(X)$ factors into factors of degree at most 2.

Notice that the degree of $f_{d_1,\dots,d_r}(X)$ equals 2^r and that the numbers of factors modulo any prime p is at least 2^{r-1}.

In the beautiful paper [4] by A.K. Lenstra, H.W. Lenstra jr. and L. Lovász, the authors published a factorisation algorithm which is indeed polynomial in the degree of f and $\log||f||$. It is based on their method of lattice reduction. Beside its application in factorisation this technique has found wide applications in many other areas of computational mathematics.

6. Factorisation of Polynomials with Integer Coefficients, II

We now sketch a factorisation algorithm in $\mathbb{Z}[X]$ based on the LLL-algorithm. Let f be the polynomial to be factored and assume that it is squarefree and primitive. Let n be its degree. Let p be a prime not dividing the discriminant. Let $h \in \mathbf{Z}[X]$ be a monic polynomial, irreducible in $\mathbf{F}_p[X]$ and suppose

h mod p divides f mod p. Our problem will be to find the primitive irreducible divisor h_0, unique up to a factor ±1, of f such that h mod p divides h_0 mod p.

Let l be the degree of h and let $m \geq l$. Choose $k \in \mathbb{N}$ to be specified later. Using Hensel lifting we choose h in such a way that h mod p^k divides f mod p^k. We shall identify the set of polynomials in $\mathbb{R}[X]$ of degree at most m with the set of its coefficient vectors, which is $\mathbb{R}^{m+1}$. Consider the lattice L of rank $m+1$ spanned by the polynomials

$$p^k X^i : \ 0 \leq i < l \qquad \text{and} \qquad hX^j : \ 0 \leq j \leq m - l.$$

Note that L is simply the set of polynomials of degree $\leq m$ which are divisible by h modulo p^k. Since h is monic, we easily verify that $d(L) = p^{kl}$. We now have a fundamental observation.

Proposition 6.1. *Let $b \in L$ satisfy*

$$p^{kl} > ||f||^m ||b||^n.$$

Then $\gcd(f, b) \neq 1$, *in particular b is divisible by h_0.*

Proof. Suppose that f and b do not have a common divisor of positive degree. Consider the lattice M generated by the polynomials $X^i f(X)$, $0 \leq i < m$ and $X^j b(X)$, $0 \leq j < n$. Since $\gcd(f, b)$ is constant, these polynomials are $\mathbb{Z}$-linear independent, hence they generate a lattice M of rank $n+m$. Note that the determinant $d(M)$ is precisely the absolute value of the resultant of f and b. By Hadamard's inequality it can be bounded by $||f||^m ||b||^n$. On the other hand, we can choose a basis $b_0, b_1, \ldots$ for M in Hermite normal form with respect to $1, X, X^2, \ldots, X^{n+m-1}$. Since every polynomial in M is divisible modulo p^k by h, we have that $b_0, b_1, \ldots, b_{l-1}$ are divisible by p^k. Hence, p^{kl} divides $d(M)$. So we get $p^{kl} \leq ||f||^m ||b||^n$, contradicting the inequality in the statement of our proposition.

The idea of the factorisation algorithm is now to choose k so large that a possible factor h_0, whose size we know by the Landau-Mignotte bound, can be considered as very small vector in L.

Proposition 6.2. *Let notation be as above. Suppose that $b_1, \ldots, b_{m+1}$ is a reduced basis for L, and that*

$$p^{kl} > 2^{mn/2} \binom{2m}{m}^{n/2} ||f||^{m+n}.$$

Then we have $\deg(h_0) \leq m$ *if and only if*

$$||b_1|| < (p^{kl}/||f||^m)^{1/n}.$$

Proof. The 'if'-part is immediate from the previous proposition. Suppose now that $\deg(h_0) \leq m$. Then L contains a vector, namely h_0, whose norm is bounded by $\binom{2m}{m}^{1/2}||f||$ (Landau-Mignotte). From the properties of a reduced basis we then know that $||b_1|| \leq 2^{m/2}||h_0|| \leq 2^{m/2}\binom{2m}{m}^{1/2}||f||$. By the lower bound for k given, this implies that $||b_1|| < (p^{kl}/||f||^m)^{1/n}$.

We observe that the above two propositions already suffice to yield a factorisation algorithm. However, there is one more result which tells us how to find h_0 without much effort after the lattice reduction has been performed.

Proposition 6.3. *Let notation be as in the previous proposition. Suppose there is an index j such that*

$$||b_j|| < (p^{kl}/||f||^m)^{1/n}.$$

Let t be the largest such j. Then

$$\deg(h_0) = m + 1 - t \quad \textit{and} \quad h_0 = \pm\gcd(b_1, \ldots, b_t).$$

Proof. Let J be the set of all j for which the inequality $||b_j|| < (p^{kl}/||f||^m)^{1/n}$ holds. We know that b_j is divisible by h_0 for every $j \in J$. Furthermore the b_j are $\mathbb{R}$-linearly independent and we have $\deg(h_0) \leq m + 1 - |J|$, hence $|J| \leq m+1-\deg(h_0)$. On the other hand we have for every i that $||X^i h_0|| = ||h_0|| \leq \binom{2m}{m}^{1/2}||f||$. For $i = 0, \ldots, m - \deg(h_0)$ the polynomials $X^i h_0$ are in L and by the properties of a reduced basis we get $||b_j|| \leq 2^{m/2}\binom{2m}{m}^{1/2}||f||$ for $j = 0, \ldots, m - \deg(h_0)$. Using the lower bound for p^{kl} we get $||b_j|| < (p^{kl}/||f||^m)^{1/n}$ for $j = 0, \ldots, m - \deg(h_0)$ and we infer that $|J| \geq m + 1 - \deg(h_0)$. Combined with the upper bound for $|J|$ this implies $J = \{1, \ldots, m + 1 - \deg(h_0)\}$. We conclude that $t = m + 1 - \deg(h_0)$ and that $\deg(h_0) = \deg(\gcd(b_1, \ldots, b_t))$.

It remains to show that $\gcd(b_1, \ldots, b_t)$ is primitive to establish $h_0 = \pm\gcd(b_1, \ldots, b_t)$. Note that b_1 is divisible by h_0. Hence $b_1/c(b_1)$ is divisible by h_0, hence $b_1/c(b_1) \in L$. Since b_1 is a basis vector of L we conclude that $c(b_1) = 1$. Hence b_1 and, a fortiori, $\gcd(b_1, \ldots, b_t)$ are primitive.

It does not take much imagination to see how one can construct a factorisation algorithm in $\mathbb{Z}[X]$ from the above propositions. In their article, Lenstra, Lenstra and Lovász carry out a running time analysis and they conclude that the number of bit operations is bounded by $O(n^{12}+n^9(\log||f||)^3)$.

7. Factorisation in $K[X]$, K Algebraic Number Field

This section requires some knowledge of algebraic number fields. Let K be an algebraic number field of degree n and let σ_i $(i = 1, \ldots, n)$ be its embeddings in $\mathbb{C}$. We also choose a primitive element $\theta \in K$, i.e., $K = \mathbb{Q}(\theta)$.

We like to factor a given polynomial $A \in K[X]$ in $K[X]$. Of course we may assume that A is squarefree. Given a polynomial $A \in K[X]$ we define its norm by

$$\mathcal{N}(A) = \prod_{i=1}^{n} \sigma_i(A).$$

Since the coefficients of $\mathcal{N}(A)$ are symmetric functions of the $\sigma_i(\theta)$ we have that $\mathcal{N}(A) \in \mathbb{Q}[X]$.

In order to perform the factorisation we need the following lemmas.

Lemma 7.1. *Let $A \in K[X]$ be irreducible. Then $\mathcal{N}(A)$ is a power of an irreducible polynomial in $\mathbb{Q}[X]$.*

Proof. Let $\prod_i N_i^{e_i}$ be the factorisation of $\mathcal{N}(A)$ into irreducible factors in $\mathbb{Q}[X]$. Since A divides this product and A is irreducible, there exists an i such that A divides N_i. Consequently $\sigma_j(A)$ divides N_i for every j and hence $\mathcal{N}(A)$ divides N_i^n. Thus our Lemma follows.

Lemma 7.2. *Let $A \in K[X]$ be a squarefree polynomial. Then for all but finitely many $k \in \mathbb{Q}$ the polynomial $\mathcal{N}(A(X - k\theta))$ is squarefree.*

Proof. We denote the zeros of $\sigma_j(A)$ by $\beta_{i,j}$ where $j = 1, \ldots, n$, $i = 1, \ldots, m$ and m is the degree of A. So the zeros of $\sigma_j(A(X - k\theta))$ are given by $\beta_{i,j} + k\sigma_j(\theta)$. The polynomial $\mathcal{N}(A)$ is not squarefree if at least two of such zeros coincide. However, it is easy to see that this can happen for at most finitely many $k \in \mathbb{Q}$.

Proposition 7.3. *Let $B \in K[X]$ be squarefree and assume that $\mathcal{N}(B)$ is squarefree. Let $\prod_i N_i$ be the irreducible factorisation of $\mathcal{N}(B)$ in $\mathbb{Q}[X]$. Then the irreducible factorisation of B in $K[X]$ is given by $\prod_i \gcd(B, N_i)$.*

Proof. Let $B_1, \ldots, B_g$ be the irreducible factors of B. We know that $\mathcal{N}(B_i)$ is a power of an irreducible polynomial for every i. But they also divide $\mathcal{N}(B)$ which is known to be squarefree. Hence the $\mathcal{N}(B_i)$ are irreducible and distinct. So, after reordering of indices we may assume $\mathcal{N}(B_i) = N_i$ for $i = 1, \ldots g$. It now follows that $B_i = \gcd(B, N_i)$, which proves our assertion.

Factorisation of a squarefree polynomial $A \in K[X]$ is now straightforward. First find $k \in \mathbb{Q}$ such that $\mathcal{N}(B)$ with $B(X) = A(X - k\theta)$ is squarefree. This will usually be the case after a very few trials for k. Determine the irreducible factors N_i of $\mathcal{N}(B)$. Then the irreducible factors of B are given by the $\gcd(B, N_i)$. The factors of A are easily recovered.

Notes

During the preparation we made good use of the very nice article [3]. In most books on computational number theory one can find a description of Berlekamp's algorithm. In particular, we mention the beautiful books [1] and [3] of which we have made good use in the preparation of these lectures. References to more recent articles are given below.

References

1. H. Cohen (1995): *A Course in Computational Algebraic Number Theory* (2nd edition), Springer-Verlag, Berlin Heidelberg New York.
2. M.F. Singer and F. Ulmer (1993): *Galois groups of second and third order linear differential equation*, J. Symbolic Computation **16**, 9–36.
3. A.K. Lenstra (1982): *Factorisation of polynomials*, in: Computational Methods in Number Theory, Mathematical Centre Tract **154** (H.W. Lenstra jr, R. Tijdeman, eds.).
4. A.K. Lenstra, H.W. Lenstra jr., and L. Lovász (1982): *Factoring polynomials with rational coefficients*, Math. Ann. **261**, 515–534.
5. M. Mignotte (1982): *Some useful bounds.* Computer algebra, symbolic and algebraic computation, Comput. Suppl. **4**, 259–263.
6. M. van Hoeij (1996): *Factorization of linear differential operators*, PhD. Thesis, Katholieke Universiteit Nijmegen, The Netherlands.

Chapter 5. Computations in Associative and Lie Algebras

Gábor Ivanyos and Lajos Rónyai

1. Introduction

In this chapter we consider some basic algorithmic problems related to finite dimensional associative algebras. Our starting point is the structure theory of these algebras. This theory gives a description of the main structural ingredients of finite dimensional associative algebras, and specifies the way the algebra is constructed from these building blocks. We describe polynomial time algorithms to find the main components: the Jacobson radical and the simple direct summands of the radical-free part.

Next we look at the problem of exploring the structure of simple algebras. On the constructive side we discuss algorithms for finding zero divisors in finite algebras. We show that this problem belongs to the same complexity class as the task of factoring polynomials over finite fields.

We touch upon some applications of the associative decomposition algorithms. These include efficient algorithms for calculating the radicals (solvable and nilpotent) of Lie algebras. We also outline a randomised polynomial time algorithm to find a common invariant subspace of a set of linear transformations over a finite field.

2. Basic Definitions and Structure Theorems

First we give some basic definitions related to associative algebras. A linear space $\mathcal{A}$ over the field F is an *algebra* over F if it is equipped with a binary, F-bilinear operation (called multiplication). It is customary to denote the product of $x, y \in \mathcal{A}$ by xy. Multiplication is assumed to be associative, i.e.,

$$x(yz) = (xy)z \text{ for every } x, y, z \in \mathcal{A}.$$

We restrict our attention to finite dimensional algebras only. We shall assume throughout that $\dim_F \mathcal{A} = n < \infty$. We say that $\mathcal{A}$ is a *commutative algebra* if $xy = yx$ for every $x, y \in \mathcal{A}$. An F-subspace $\mathcal{B}$ of $\mathcal{A}$ is a *subalgebra* of $\mathcal{A}$, if $\mathcal{B}$ is closed under multiplication: if $x, y \in \mathcal{B}$, then $xy \in \mathcal{B}$. An important example of a subalgebra is the *centre* $C(\mathcal{A}) = \{x \in \mathcal{A} \mid xy = yx \text{ for every } y \in \mathcal{A}\}$. If $\mathcal{A}$ has a multiplicative identity element $1 = 1_{\mathcal{A}}$, then we have $F = F \cdot 1_{\mathcal{A}} \leq C(\mathcal{A})$ (here $\leq$ is used to denote a subalgebra). The algebra $\mathcal{A}$ is *central* (over F) if it has an identity element $1 = 1_{\mathcal{A}}$ and $C(\mathcal{A}) = F$. An

F-subspace $\mathcal{I}$ of $\mathcal{A}$ is a *left ideal* of $\mathcal{A}$ if $yx \in \mathcal{I}$ whenever $x \in \mathcal{I}$ and $y \in \mathcal{A}$. A *right ideal* is defined analogously. An F-subspace $\mathcal{I}$ of $\mathcal{A}$ is an *ideal* of $\mathcal{A}$ if $\mathcal{I}$ is both left and right ideal of $\mathcal{A}$. If $\mathcal{I}$ is an ideal in $\mathcal{A}$, then we can form the *factor algebra* $\mathcal{A}/\mathcal{I}$. The notions of *homomorphism* and $\mathcal{A}$-*module* are used in the standard way, cf. Herstein [16], Pierce [26].

A couple of elements $0 \neq x, y \in \mathcal{A}$ is a pair of *zero divisors* in $\mathcal{A}$, if $xy = 0$. A nonzero element $e \in \mathcal{A}$ is an *idempotent* if $e^2 = e$.

The algebra $\mathcal{A}$ is *simple* if it has no ideals except (0) and $\mathcal{A}$, and $\mathcal{A}\mathcal{A} \neq (0)$, where $\mathcal{A}\mathcal{A}$ is the algebra generated by products ab with $a, b \in \mathcal{A}$. We say that $\mathcal{A}$ is the direct sum of its (left) ideals $\mathcal{A}_1, \ldots, \mathcal{A}_k$ (written as $\mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_k$) if $\mathcal{A}$ is the direct sum of these linear subspaces.

Examples 2.1. (1) Let the field K be a finite algebraic extension of F. In this case K is a simple and commutative (associative) algebra over F.

(2) $\mathrm{M}_d(F)$, the algebra of all d by d matrices over F. Here multiplication is the familiar matrix multiplication. It is not difficult to see that $\mathrm{M}_d(F)$ is a simple algebra over F.

(3) Let G be a (multiplicatively written) group. The elements $g \in G$ form an F-basis of the *group algebra* $F[G]$. The elements of $F[G]$ are linear combinations $\sum_i \alpha_i g_i$ with $\alpha_i \in F$ and $g_i \in G$. Multiplication is defined by $(\sum_i \alpha_i g_i) \cdot (\sum_j \beta_j g_j) = \sum_i \sum_j \alpha_i \beta_j g_i g_j$. From the associativity of the group law it follows that $F[G]$ is an associative algebra.

(4) Subalgebras of $\mathrm{M}_d(F)$. These are linear subspaces of $\mathrm{M}_d(F)$ closed with respect to matrix multiplication.

The latter examples are in a sense quite general, as the following straightforward fact shows.

Representation Theorem 2.2. *Let $\mathcal{A}$ be an algebra over the field F and suppose that* $\dim_F \mathcal{A} = n$. *Then $\mathcal{A}$ is isomorphic to a subalgebra of* $\mathrm{M}_{n+1}(F)$.

Proof. We shall use the *regular representation*. For $x \in \mathcal{A}$ we define the linear map $R_x : \mathcal{A} \to \mathcal{A}$ as $R_x(y) = xy$ for every $y \in \mathcal{A}$. It is immediate that R is a homomorphism of $\mathcal{A}$ to the algebra of linear transformations of the F-space $\mathcal{A}$. If $\mathcal{A}$ has an identity element, then R is injective, hence we have an embedding of $\mathcal{A}$ into $\mathrm{M}_n(F)$. If $\mathcal{A}$ has no identity element, then first we adjoin one using the Dorroh extension (Exercise 2.7). This increases the dimension by one and now the regular representation embeds $\mathcal{A}$ into $\mathrm{M}_{n+1}(F)$.

An element $x \in \mathcal{A}$ is called *nilpotent* if $x^k = 0$ for some positive integer k (which may depend on x). An element $x \in \mathcal{A}$ is *strongly nilpotent* if xy is nilpotent for every $y \in \mathcal{A}$. The *Jacobson radical* $\mathrm{Rad}(\mathcal{A})$ of $\mathcal{A}$ is the set of strongly nilpotent elements of $\mathcal{A}$. It is not difficult to see that $\mathrm{Rad}(\mathcal{A})$ is an ideal of $\mathcal{A}$ (Exercises 3.9–3.11) and that the factor $\mathcal{A}/\mathrm{Rad}(\mathcal{A})$ has no nonzero strongly nilpotent elements. It can be shown also, that $\mathrm{Rad}(\mathcal{A})$ is a *nilpotent ideal*: there exists a positive integer k such that $x_1x_2\cdots x_k = 0$, for all $x_1, x_2, \ldots, x_k \in \mathrm{Rad}(\mathcal{A})$.

An algebra $\mathcal{A}$ is *semisimple* if $|\mathcal{A}| \geq 2$ and $\mathrm{Rad}(\mathcal{A}) = (0)$. There is a very strong and useful characterization of semisimple algebras (Herstein [16], Pierce [26]):

Wedderburn's Theorem 2.3. *If $\mathcal{A}$ is a finite dimensional semisimple associative algebra over the field F, then $\mathcal{A}$ is expressible as a direct sum*

$$\mathcal{A} = \mathcal{A}_1 \oplus \mathcal{A}_2 \oplus \cdots \oplus \mathcal{A}_k, \tag{2.1}$$

where the $\mathcal{A}_i$ are exactly the minimal nonzero ideals of $\mathcal{A}$. Moreover, $\mathcal{A}_i$ is isomorphic to a full (and simple) matrix algebra $\mathrm{M}_{n_i}(F_i)$ where F_i is a possibly noncommutative extension field of F $(1 \leq i \leq k)$.

Another theorem by Wedderburn (Herstein [16], p. 71) implies that if F is finite then the fields F_i are actually commutative.

Example 2.4. Let $\mathcal{S}_3 \leq \mathrm{M}_3(F)$ be the subspace of the matrices with all column and row sums equal. It is immediate that if $a \in \mathcal{S}_3$ with column and row sums α, and $b \in \mathcal{S}_3$ with column and row sums β, then all row and column sums of ab are $\alpha\beta$. This implies that $\mathcal{S}_3$ is actually a subalgebra of $\mathrm{M}_3(F)$.

Exercise 2.5. Prove that $\dim_F \mathcal{S}_3 = 5$. (Hint: There is exactly one matrix in $\mathcal{S}_3$ with arbitrarily specified entries in the first row and in the first two positions of the second row.)

Now we show that $\mathcal{S}_3$ is a semisimple algebra over F. In fact, we proceed directly to determine the Wedderburn decomposition of $\mathcal{S}_3$ into simple algebras. Let F^3 denote the F-space of column vectors of length 3 and let $\mathrm{e}_i \in F^3$ stand for the column vector with 1 in the i-th position and 0 elsewhere $(i = 1, 2, 3)$. Let $U \leq F^3$ denote the subspace spanned by the vector $\mathrm{e}_1 + \mathrm{e}_2 + \mathrm{e}_3$ and $U' \leq F^3$ the subspace of all vectors whose components sum up to 0. Straightforward calculation verifies that $F^3 = U \oplus U'$ and that $\mathcal{S}_3$ leaves U and U' invariant: if $v \in U$, $w \in U'$ and $a \in \mathcal{S}_3$ then $av \in U$ and $aw \in U'$. Let $\mathcal{B} \leq \mathrm{M}_3(F)$ be the subalgebra of *all* matrices b for which $bU \subseteq U$ and $bU' \subseteq U'$ hold. We have $\mathcal{B} \cong F \oplus \mathrm{M}_2(F)$, hence

$$\dim_F \mathcal{B} = \dim_F F + \dim_F \mathrm{M}_2(F) = 1 + 4 = 5.$$

Now $\mathcal{S}_3 \leq \mathcal{B}$ together with Exercise 2.5 gives that $\mathcal{B} = \mathcal{S}_3$.

Next we explore the structure of the group algebra $\mathbb{F}_2[\mathrm{S}_3]$ of the symmetric group S_3 on the three letters 1,2,3. The base field is $\mathbb{F}_2$, the two-element field. Let $x \in \mathbb{F}_2[\mathrm{S}_3]$ be the sum of the elements of S_3: $x = \sum_{\pi \in \mathrm{S}_3} \pi$. We have $\pi x = x\pi = x$ for every $\pi \in \mathrm{S}_3$. This implies that the subspace $\mathcal{I}$ spanned by x is an ideal of $\mathbb{F}_2[\mathrm{S}_3]$. This is a nilpotent ideal because $x^2 = 6x = 0$. We infer that $\mathcal{I} \leq \mathrm{Rad}(\mathbb{F}_2[\mathrm{S}_3])$. We show that we have equality here; $\mathcal{I}$ is, in fact, the radical of the group algebra.

To this end we define a homomorphism Φ of S_3 into the group of invertible matrices from $\mathrm{M}_3(\mathbb{F}_2)$. The image $\Phi(\pi)$ of the permutation π of $\{1,2,3\}$ is the linear map which sends e_i to $\mathrm{e}_{\pi(i)}$. For example

$$\Phi((123)) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{ and } \Phi((23)) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The map Φ extends to an algebra homomorphism, which we also denote by Φ, of $\mathbb{F}_2[\mathrm{S}_3]$ into $\mathrm{M}_3(\mathbb{F}_2)$, actually into $\mathcal{S}_3$. The ideal $\mathcal{I}$ is in the kernel of Φ because $\Phi(x) = \sum_{\pi \in \mathrm{S}_3} \Phi(\pi) = 0$.

Exercise 2.6. Verify that $\Phi(\mathbb{F}_2[\mathrm{S}_3]) = \mathcal{S}_3$. (Hint: Check that the matrices $\Phi(\pi)$, $\pi \in \mathrm{S}_3$, $\pi \neq id$ are linearly independent over $\mathbb{F}_2$.)

From the exercise we see (comparing dimensions) that the kernel of Φ is $\mathcal{I}$. The image is a semisimple algebra, hence $\mathrm{Rad}(\mathbb{F}_2[\mathrm{S}_3]) \leq \mathcal{I}$. This, with the reverse inclusion gives that $\mathrm{Rad}(\mathbb{F}_2[\mathrm{S}_3]) = \mathcal{I}$.

We have thus determined the major structural building blocks of the group algebra $\mathbb{F}_2[\mathrm{S}_3]$. It has a one-dimensional radical; the semisimple factor splits into two simple components.

In this chapter we discuss algorithms related to the structure theory outlined in the preceding paragraphs. We explain some basic methods for finding in a computationally efficient way the structural ingredients of algebras. In Section 3 we present algorithms for computing the Jacobson radical. Here the interesting case is when F (and consequently $\mathcal{A}$) is finite. These methods are applied in Section 4 to the computation of the (solvable) radical and the nilradical of Lie algebras.

The computation of the Wedderburn decomposition (2.1) of semisimple associative algebras is the subject of Section 5. We explain a solution for the case $|\mathcal{A}| < \infty$, and outline extensions to the infinite case, where the main concern is controlling the size of the results.

In Section 6 an algorithm is presented for finding zero divisors in a finite algebra $\mathcal{A}$: nonzero elements $x, y \in \mathcal{A}$ such that $xy = 0$. It may come as a surprise, that the case $\mathcal{A} = \mathrm{M}_n(F)$ causes most of the complications here. The infinite version of the problem looks much more difficult. Some evidence pointing to this is given in Project 4.

Some of the exercises include algebraic facts which are related to our subject matter. Others are devoted to algorithmic topics. Highlights are the applications of the zero divisor algorithm. For example, the method can be used to find a common invariant subspace for a set of linear operators over a finite field.

Motivation for considering algorithms for associative algebras can be drawn from a very basic mathematical principle. In theoretical considerations as well as in practical applications one frequently has some linear operators $X_1, X_2, \ldots, X_k \in \mathrm{M}_n(F)$ and is interested in the common invariant subspaces of these operators. These subspaces are precisely the common invariant subspaces of $\mathcal{A}$, the algebra generated by the X_i. In studying these subspaces, one may often take advantage of the richer structure of $\mathcal{A}$.

As a simple example of this, we refer back to Example 2.4. We can gain valuable information about the action on the space ${\mathbb{F}_2}^3$ of the matrices $\Phi(\pi)$, $\pi \in \mathrm{S}_3$ by working with the algebra $\mathcal{S}_3$ generated by them.

We are interested in exact (symbolic) computations. For this reason we consider ground fields admitting efficient exact arithmetic. In this chapter F will be either a finite field or an algebraic number field.

We specify now the input of the algorithmic problems addressed. To obtain sufficiently general results, we consider an algebra to be given as a collection of structure constants. If $\mathcal{A}$ is an algebra over the field F and $a_1, a_2, \ldots, a_n$ is a basis of the F-space $\mathcal{A}$, then multiplication is completely described if we express the products $a_i a_j$ as linear combinations of the basis elements:

$$a_i a_j = \gamma_{ij1} a_1 + \cdots + \gamma_{ijn} a_n.$$

The coefficients $\gamma_{ijk} \in F$ are called *structure constants.* When an algebra is given as input, we assume that it is represented as an array of structure constants. Substructures (such as subalgebras, ideals, subspaces) can then be represented by bases whose elements are linear combinations of basis elements of the ambient structure (algebra).

In our cases F can be viewed as an algebra over its prime field P; therefore F can also be represented with structure constants from P. (If F is finite then $P = \mathbb{F}_p$ for some prime p, if F is a number field then $P = \mathbb{Q}$.) In these cases F is usually specified by giving the (monic) minimal polynomial f of a single generating element α over the prime field P. This is a special case of the representation with structure constants. The coefficients of f give the structure constants with respect to the P-basis $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$ of F where $n = \dim_P F$.

Another important way to represent an algebra is in the form of a matrix algebra. In these cases we are given a collection of matrices which generate the algebra. The algorithms described in these notes are applicable in this setting as well. From such a matrix representation one can efficiently find a basis of the algebra and then calculate structure constants with respect to this basis (see Exercise 2.8).

We would like to consider algorithms which have a theoretical guarantee for their efficiency. From the perspective of computer science these are the polynomial time algorithms. An algorithm runs in polynomial time if, on inputs of length n (n is a positive integer) the computation requires at most n^c bit-operations. Here $c > 0$ is a constant independent of n. We refer to [17], Chapters 12–13 and [8] for the basic notions related to the complexity of algorithms.

The length (or size) of the objects we work with is defined in quite a natural way. The size of a natural number is the number of bits in its binary representation. That is, the size of m is $\lceil \log_2(m+1) \rceil$. The size of a rational number p/q expressed in lowest terms is size(p)+size(q). Modulo p residue classes have size $\lceil \log_2(p+1) \rceil$. The size of objects built up from simpler ones (polynomials, vectors, matrices, arrays of structure constants, etc.) is the sum of the sizes of their components.

We now briefly comment on the most important algorithmic tools employed in the methods to be described later on. Polynomial time procedures are available for the fundamental seminumerical computations (such as the arithmetical operations in F and polynomial arithmetic over F), if F is a finite field or an algebraic number field. The reader is referred to Collins, Mignotte, and Winkler [7], Knuth [22] for the details. The basic algorithmic tasks of linear algebra (such as testing linear dependence, computing ranks, determinants, and solving systems of linear equations) can be implemented in deterministic polynomial time over the fields considered. If F is finite, then the well-known textbook methods demonstrate this point. In the number-field-case it will be sufficient to solve linear algebra problems over the field of rationals $\mathbb{Q}$. There are polynomial time methods to solve systems of linear equations over $\mathbb{Q}$ and over the integers $\mathbb{Z}$ [1], [12], [21].

The algorithmic problem of factoring polynomials over F turns out to be very important for us. We refer the reader to Chapter 4 for an introduction to this subject. We record here only that there are deterministic polynomial time methods to solve such problems over number fields.

The picture is a bit more complicated when the ground field is finite, say $F = \mathbb{F}_q$. There are deterministic methods – the first one given by Berlekamp [3], [24] – with time complexity polynomial in the parameters p, s and $\deg(f)$ where $f(X) \in \mathbb{F}_q[X]$ is the polynomial to be factored and $q = p^s$, p prime. Note that the input size in this case is about $(1+\deg(f))\log q$, consequently the running time of the method is not bounded by a polynomial in the input size. The problem can be solved in polynomial time if we allow randomization. The first such method was also proposed by Berlekamp [4]. The method belongs to a special kind of randomised methods, the so-called *Las Vegas algorithms*. A Las Vegas algorithm for an arbitrary input either gives a correct solution or, with small probability, admits failure. The point is, that a method of this kind never gives a misleading answer.

As we have already suggested, some of the algorithms we are to discuss require factoring polynomials over finite fields. We intend to make clear the dependence of the methods on this algorithmic ingredient. For this reason, we define a deterministic algorithm to be an *f-algorithm*, if it is allowed to call an oracle (subroutine or procedure) for factoring polynomials over finite fields. The cost of a call is the size of the input passed on to the procedure.

A polynomial time f-algorithm can be considered as practical, because the factoring oracle admits Las Vegas polynomial time implementations.

For the next exercises $\mathcal{A}$ denotes a finite dimensional associative algebra over the field F.

Exercise 2.7. We define the *Dorroh extension* $\mathcal{A}^*$ of $\mathcal{A}$ as the set of pairs (a, λ), where $a \in \mathcal{A}$ and $\lambda \in F$. Addition and multiplication on $\mathcal{A}^*$ are defined as follows:

$$(a_1, \lambda_1) + (a_2, \lambda_2) = (a_1 + a_2, \lambda_1 + \lambda_2),$$

$$(a_1, \lambda_1)(a_2, \lambda_2) = (a_1 a_2 + \lambda_1 a_2 + \lambda_2 a_1, \lambda_1 \lambda_2).$$

Show that $\mathcal{A}^*$ is an F-algebra with an identity element. Moreover $\mathcal{A}^*$ contains a subalgebra (in fact, an ideal) isomorphic to $\mathcal{A}$.

Suppose that $\mathcal{A}$ is given as a collection of structure constants $\gamma_{ijk} \in F$ with respect to an F-basis $a_1, a_2, \ldots, a_n$.

Exercise 2.8. Let $b_1, \ldots, b_k$ be elements of $\mathcal{A}$ given as linear combinations of the a_i. Give an algorithm which employs at most $O(n^4)$ arithmetical operations over F to compute an F-basis of the subalgebra (ideal) generated by the b_j. (Hint: Use the fact that a system of linear equations over F with n variables and n equations can be solved with $O(n^3)$ arithmetical operations.)

Exercise 2.9. Suppose that we are given bases of two F-subspaces U, V of $\mathcal{A}$. Find a basis for $U \cap V$ at the expense of $O(n^3)$ arithmetical operations over F. (Hint: Find first dual bases for U and V.)

3. Computing the Radical

Here we consider the problem of computing (a basis of) the radical $\mathrm{Rad}(\mathcal{A})$ of an algebra $\mathcal{A}$. We give polynomial time algorithms over the ground fields we work with. The main algorithmic problem we address is the following.

Suppose $\mathcal{A}$ is a finite dimensional associative algebra over the field F, given as a collection of structure constants. Our objective is to find a basis of $\mathrm{Rad}(\mathcal{A})$, the radical of $\mathcal{A}$, in time polynomial in the input size.

If $\mathrm{char}\, F = 0$, then the problem is equivalent to solving a system of linear equations over the ground field as follows from the characterization of the radical by Dickson [9], pp. 106–108:

Dickson's Theorem 3.1. *Let $\mathcal{A}$ be a finite dimensional algebra of matrices over a field F, and* $\mathrm{char}\, F = 0$. *Then*

$$\mathrm{Rad}(\mathcal{A}) = \{x \in \mathcal{A} \mid \ \mathrm{Tr}(yx) = 0 \text{ for every } y \in \mathcal{A}\}.$$

In fact, if $a_1, \ldots, a_n$ is a linear basis of $\mathcal{A}$ over F, then to find $\mathrm{Rad}(\mathcal{A})$, it suffices to solve the linear system $\mathrm{Tr}(a_i x) = 0$, $i = 1, \ldots, n$, where x is an 'unknown' element of $\mathcal{A}$.

We now turn to the case where $\mathcal{A}$ (and hence $F = \mathbb{F}_q$) is finite. We assume that p is a prime, q is a power of p, $F = \mathbb{F}_q$ and that $\mathcal{A}$ is a subalgebra of

$\mathrm{M}_n(F)$. Using the regular representation of $\mathcal{A}$ (or that of its Dorroh extension, if necessary), we can efficiently achieve this situation. The statement of Dickson's Theorem is no longer valid in positive characteristic (see Exercise 3.13). There is, however, a more subtle, and still useful, description of the radical in this case. We explain this in the sequel.

We define the natural number l by the following inequalities: $p^l \leq n < p^{l+1}$. Let B denote the set of matrices $\mathcal{A} \cup \{I\}$ where I is the identity element of $\mathrm{M}_n(F)$. Let $a \in \mathrm{M}_n(F)$ be a matrix. It will be convenient to work with the following variant of the characteristic polynomial of a:

$$\widetilde{\chi}_a(X) = \det(Xa + I) \in F[X].$$

Consider the expansion of $\widetilde{\chi}_a(X)$ as a polynomial in the variable X:

$$\widetilde{\chi}_a(X) = 1 + \sum_{i=1}^{n} c_{a,i} X^i.$$

The indices of the form $i = p^j$, $j = 0, \ldots, l$ play a key role in the following arguments. For $j = 0, \ldots, l$ we define the 'trace functions' T_j by

$$T_j(a) := c_{a,p^j}.$$

Obviously, $T_0(a) = \mathrm{Tr}(a)$ is the (ordinary) trace of the matrix a.

We also define a sequence $\mathcal{A} =: \mathcal{R}_0 \supseteq \mathcal{R}_1 \supseteq \cdots \supseteq \mathcal{R}_{l+1}$ of subsets of $\mathcal{A}$ as

$$\mathcal{R}_j := \{a \in \mathcal{A} \mid T_i(ba) = 0 \text{ for every } b \in B \text{ and } 0 \leq i < j\} \quad (1 \leq j \leq l+1).$$

Alternatively, for every $0 \leq j \leq l$, we have

$$\mathcal{R}_{j+1} := \{a \in \mathcal{R}_j \mid T_j(ba) = 0 \text{ for every } b \in B\}.$$

At this point we can formulate a characterization of $\mathrm{Rad}(\mathcal{A})$ which is useful for computational purposes. It is the main result of this section and reads as follows:

Theorem 3.2. *Let $\mathcal{A} \leq \mathrm{M}_n(F)$ be an algebra of matrices over the finite field F of characteristic p. Put $l = \lfloor \log_p n \rfloor$, and let $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_{l+1}$ be as defined above. Then*

1. *$\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_{l+1}$ are ideals of $\mathcal{A}$;*
2. *$\mathcal{R}_{l+1} = \mathrm{Rad}(\mathcal{A})$;*
3. *For every $j \in \{0, \ldots, l\}$ the function T_j is p^j-semilinear on $\mathcal{R}_j$, i.e.,*

$$T_j(\alpha a + \beta b) = \alpha^{p^j} T_j(a) + \beta^{p^j} T_j(b)$$

for every $\alpha, \beta \in F$ and $a, b \in \mathcal{R}_j$.

Property 3 implies that we can obtain a basis of $\mathcal{R}_{j+1}$ from a basis of $\mathcal{R}_j$ by solving a system of linear equations over F. Indeed, set $a_0 = I$, and let $a_1, \ldots, a_d$ be a basis of $\mathcal{A}$ over F. Suppose that we have a basis $\{h_1, \ldots, h_r\}$

of $\mathcal{R}_j$ over F, and we are looking for a basis of $\mathcal{R}_{j+1}$. Semilinearity implies that an element $a \in \mathcal{R}_j$, $a = \sum_{i=1}^r \lambda_i h_i$ is in $\mathcal{R}_{j+1}$ if and only if

$$\sum_{i=1}^{r} T_j(a_t h_i)\lambda_i^{p^j} = 0, \quad (t = 0, \ldots, d).$$

The inverse of the automorphism $\lambda \mapsto \lambda^{p^j}$ of the finite field $F = \mathbb{F}_q$ can be computed efficiently (Exercise 5.6), hence the system above can be efficiently translated into

$$\sum_{i=1}^{r} T_j(a_t h_i)^{\frac{1}{p^j}} \lambda_i = 0 \quad (t = 0, \ldots, d).$$

This latter is a system of linear equations in the variables $\lambda_1, \ldots, \lambda_r$. Thus, we start with $\mathcal{R}_0 = \mathcal{A}$ and then in turn proceed to compute $\mathcal{R}_1, \ldots, \mathcal{R}_{l+1}$.

From a basis of $\mathcal{R}_i$ we obtain a basis of $\mathcal{R}_{i+1}$ by solving a system of linear equations over F. The number of equations and the number of variables is at most n^2, hence the system can be solved in time $(n + \log q)^{O(1)}$. We obtain a basis of $\mathrm{Rad}(\mathcal{A})$ in $l + 1 = O(\log n)$ such rounds; therefore the overall cost of the computation is $(n + \log q)^{O(1)}$ bit operations. Below we give a formal description of the algorithm.

```
$Radical(\mathcal{A})$ :=
        $A := \{I\} \cup basis\ of\ \mathcal{A}$;
        $H := basis\ of\ \mathcal{A}$;
        for $j$ from 1 to $\lfloor \log_p n \rfloor + 1$ do
          if $H \neq \emptyset$ then
            $G := \left(T_j(ah)^{\frac{1}{p^j}}\right)_{a \in A, h \in H}$
            $\Lambda := a\ basis\ of\ \ker G$;
            $H := \{\lambda_1 h_1 + \ldots + \lambda_r h_r \mid (\lambda_1, \ldots, \Lambda_r) \in \Lambda\}$;
          fi
        od
        return $H$.
```

We recall some basic concepts and facts from the representation theory of algebras which we shall use in the proof of the theorem. Let $\mathcal{A}$ be an arbitrary finite dimensional algebra over the field F. A (finite dimensional) $\mathcal{A}$-module is a finite dimensional linear space V over F equipped with a bilinear map from $\mathcal{A} \times V$ to V. As customary, we denote the image of $(a, v) \in \mathcal{A} \times V$ by av. We require that $(ab)v = a(bv)$ for every $a, b \in \mathcal{A}$ and $v \in V$. As an important example, the matrices from $\mathrm{M}_n(F)$ act on the linear space F^n of column vectors of length n in the natural way (multiplication from the left). It is easy to see that this multiplication defines on F^n an $\mathrm{M}_n(F)$-module structure (and an $\mathcal{A}$-module structure as well for every subalgebra $\mathcal{A} \leq \mathrm{M}_n(F)$).

The preceding construction has a (partial) converse. Let V be an $\mathcal{A}$-module with $\dim_F V = n$. We fix an F-basis of V and for every $a \in \mathcal{A}$ we take $\psi(a) \in \mathrm{M}_n(F)$ which is the matrix of the linear map $v \mapsto av$, called the *action* of a on V. It is easy to check that the map $\psi : \mathcal{A} \to \mathrm{M}_n(V)$ is an algebra homomorphism. This map is called the *matrix representation* of $\mathcal{A}$ corresponding to the module V. Of course, ψ is defined only up to change of bases in V. An *$\mathcal{A}$-module isomorphism* of two $\mathcal{A}$-modules V and W is an F-linear bijection $\phi : V \to W$ satisfying $\phi(av) = a\phi(v)$ for every $a \in \mathcal{A}$ and $v \in V$.

In investigations related to substructures of an $\mathcal{A}$-module V it is convenient to use the notation HU where $H \subseteq \mathcal{A}$ and $U \subseteq V$. The subspace $HU \leq V$ is the linear span of the elements hu, where $h \in H$ and $u \in U$. A *submodule* (or an $\mathcal{A}$-invariant subspace) of V is an F-linear subspace $U \subseteq V$ such that $\mathcal{A}U \leq U$, i.e., $au \in U$ for every $a \in \mathcal{A}$ and $u \in U$. Restriction of the action of $\mathcal{A}$ to a submodule U makes U obviously an $\mathcal{A}$-module. Also $\mathcal{A}$ acts on the factor space V/U in a natural way: $a(v+U) := av + U$. It is easy to see that this multiplication makes the factor space V/U an $\mathcal{A}$-module, called the *factor module.*

Obviously, (0) and V are always submodules. Modules V with exactly two submodules are of particular interest. If $\mathcal{A}V = (0)$ then V is called a zero module. Since the action of $\mathcal{A}$ is identically zero, every subspace is a submodule; therefore $\dim_F V$ must be one. A module V with nontrivial multiplication that has exactly two submodules is called a *simple* or *irreducible* $\mathcal{A}$-module. Note that since the subspace $\mathcal{A}V$ is always a submodule, for a simple module V we have $\mathcal{A}V = V$.

By a *composition series* of V we mean a chain $(0) = V_0 < V_1 < \ldots < V_r = V$ of submodules such that for $i = 1, \ldots, r$ there exists no submodule U with $V_{i-1} < U < V_i$. In other words, the factor modules V_i/V_{i-1} are either simple modules or one-dimensional zero modules. Let $(0) = V_0 < V_1 < \ldots < V_r = V$ be a composition series of the $\mathcal{A}$-module V. Let W be a simple $\mathcal{A}$-module. The *multiplicity* of W in the composition series $(0) = V_0 < V_1 < \ldots < V_r = V$ is the number of factors V_i/V_{i-1} isomorphic to W.

Assume that W is a simple $\mathcal{A}$-module. Let J be an arbitrary (left) ideal of $\mathcal{A}$.

From the fact that the space JW is an $\mathcal{A}$-submodule, we infer that $JW = W$ or $JW = (0)$. In particular, since the radical is a nilpotent ideal, $\mathrm{Rad}(\mathcal{A})W = 0$ and $\mathcal{A} \neq \mathrm{Rad}(\mathcal{A})$. For elements $a \in \mathcal{A}$ and subsets $H \subseteq \mathcal{A}$, we denote their images under the natural map $\mathcal{A} \to \overline{\mathcal{A}} = \mathcal{A}/\mathrm{Rad}(\mathcal{A})$ by $\overline{a}$ and $\overline{H}$, respectively. The preceding observation implies that for every $a \in \mathcal{A}$, $b \in \mathrm{Rad}(\mathcal{A})$ and $v \in W$ we have $(a+b)v = av$. In other words, the action of a on W depends only on $\overline{a}$. This means that W can be considered as an $\overline{\mathcal{A}}$-module in a natural way.

Let $J_1, \ldots, J_t$ be the minimal elements of the set of ideals of $\mathcal{A}$ properly containing the radical. Wedderburn's Theorem applied to $\mathcal{A}/\mathrm{Rad}(\mathcal{A})$ shows that this set of ideals is finite, as the Wedderburn decomposition of $\overline{\mathcal{A}}$ is $\overline{J}_1 \oplus \ldots \oplus \overline{J}_t$. There exists a unique ideal $J_W \in \{J_1, \ldots, J_t\}$ such that $J_W W = W$. Existence can be seen from $W = \mathcal{A}W = \sum_{i=1}^{t} J_i W$. Uniqueness follows at once from the relations $J_i J_j \leq \mathrm{Rad}(\mathcal{A})$ for $i \neq j$. We say that the simple module W *belongs to* the ideal $J \in \{J_1, \ldots, J_t\}$ if $W = JW$.

The simple modules which belong to the same ideal $J \in \{J_1, \ldots, J_t\}$ are isomorphic. To see this, let L_J be an arbitrary left ideal $L_J \leq J$ of $\mathcal{A}$ which is minimal among the left ideals properly containing $\mathrm{Rad}(\mathcal{A})$. The image $\overline{L}_J$ of L_J is a minimal nonzero left ideal of $\overline{\mathcal{A}}$. The left ideal L_J can be considered as an $\mathcal{A}$-module in a natural way (via the multiplication in $\mathcal{A}$). Then $\mathrm{Rad}(\mathcal{A})$ is a submodule of L_J and the factor module is $\overline{L}_J$. Assume that the simple $\mathcal{A}$-module W belongs to the ideal $J \in \{J_1, \ldots, J_t\}$. We can exhibit an isomorphism $\phi : \overline{L}_J \to W$ as follows. It is straightforward to verify that the set $N = \{\overline{a} \in \overline{J} | \overline{a}W = (0)\}$ is a two-sided ideal of $\overline{J}$. From the facts that $\overline{J}$ is a simple algebra and $\overline{J}W \neq (0)$ we infer that $N = (0)$. This implies the existence of an element $v \in W$ such that $\overline{L}_J v \neq (0)$. We define the map $\phi : \overline{L}_J \to W$ as $\phi(x) := xv$. Obviously ϕ is a linear map, and ϕ commutes with the action of every element of $\mathcal{A}$: $\phi(ax) = (ax)v = a(xv) = a\phi(x)$. This means that ϕ is an $\mathcal{A}$-module homomorphism. It remains to show that ϕ is one to one. It is straightforward to see that $\ker\phi$ is a left ideal of $\overline{\mathcal{A}}$, and $\mathrm{im}\,\phi$ is a submodule of W. By construction, we have $\mathrm{im}\,\phi \geq \overline{L}_J v > (0)$. Because of minimality of $\overline{L}_J$ and irreducibility of W this is only possible if $\ker\phi = (0)$ and $\mathrm{im}\,\phi = W$.

After these preparations we prove Theorem 3.2 in a sequence of lemmas. The statement of the first lemma can be considered as a special case of the theorem, where the underlying module is simple.

Lemma 3.3. *Let J be a simple algebra over the finite field F and W be a simple J-module. Then there exists an element $a \in J$ with $\mathrm{Tr}_W(a) = 1$, where $\mathrm{Tr}_W(a)$ stands for the (ordinary) trace of the action of a on W.*

Proof. By Wedderburn's Theorem $J \cong \mathrm{M}_d(F')$, where d is a positive integer and F' is a finite extension of F. We identify J with this matrix algebra. The space ${F'}^d$ of column vectors of length d over F' forms a simple J-module. Since the simple J-modules are isomorphic, we can identify W with this module. It is easy to see that for every $a \in J$

$$\mathrm{Tr}_W(a) = \mathrm{Tr}_{F'/F}(\mathrm{Tr}(a)),$$

where the inner trace on the right hand side is the trace of a as a matrix from $\mathrm{M}_d(F')$, while the outer trace is the trace function of the field extension F'/F (or, equivalently, the trace of the regular representation of the F-algebra F'). Since finite fields are perfect, the function $\mathrm{Tr}_{F'/F}$ is nontrivial: there exists an element $\alpha \in F'$ with trace 1. The element $a \in J$ of the form

$$\begin{pmatrix} \alpha & 0 & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 \end{pmatrix}$$

has trace 1, as required. This finishes the proof of the lemma.

Below we show that semilinearity and other useful properties hold for the trace functions T_j on certain ideals.

Lemma 3.4. *Let $\mathcal{A} \leq \mathrm{M}_n(F)$ be a matrix algebra over the field F of characteristic p. Assume that $\mathcal{A} \neq \mathrm{Rad}(\mathcal{A})$. Let $(0) = V_0 < V_1 < \ldots < V_r = V$ be a composition series of the $\mathcal{A}$-module $V = F^n$. Let $J_1, \ldots, J_t$ be the minimal elements of the set of ideals of $\mathcal{A}$ properly containing $\mathrm{Rad}(\mathcal{A})$. For every index $i \in \{1, \ldots, t\}$ fix a simple $\mathcal{A}$-module W_i that belongs to the ideal J_i and denote the multiplicity of W_i in the composition series by m_i. Put $l = \lfloor \log_p n \rfloor$ and define the ideals $\mathcal{R}'_0, \mathcal{R}'_1, \ldots, \mathcal{R}'_{l+1}$ as*

$$\mathcal{R}'_j = \mathrm{Rad}(\mathcal{A}) + \sum_{p^j | m_i} J_i. \tag{3.1}$$

Then
1. $\mathcal{R}'_{l+1} = \mathrm{Rad}(\mathcal{A})$;
2. For every $j \in \{0, \ldots, l\}$ the function T_j is p^j-semilinear on $\mathcal{R}'_j$ (in the sense of Theorem 3.2).
3. $T_j(ab) = T_j(ba)$ for every $j \in \{0, \ldots, l\}$, $b \in \mathcal{A}$ and $a \in \mathcal{R}'_j$.
4. T_j is identically zero on $\mathcal{R}'_{j+1}$ $(j = 0, \ldots, l)$.
5. T_j is not identically zero on ideals J_i such that the multiplicity m_i is divisible by p^j but not by p^{j+1} $(j = 0, \ldots, l)$.

Proof. Obviously $\mathcal{R}'_j$ is an ideal of $\mathcal{A}$ containing the radical. On the other hand, from $p^{l+1} > n \geq r \geq m_i$ we infer that $\mathcal{R}'_{l+1} = \mathrm{Rad}(\mathcal{A})$. (Recall that r is the length of a composition chain of V.)

For an arbitrary (finite dimensional) $\mathcal{A}$-module U and $a \in \mathcal{A}$ let $\widetilde{\chi}_{U,a}(X)$ denote the variant $\widetilde{\chi}_{\psi(a)}(X) = \det(X\psi(a) + I_U)$ of the characteristic polynomial. Here, $\psi(a)$ stands for the matrix of the action of a on U written in terms of a basis of U and I_U is the appropriate identity matrix. Although ψ is defined only up to a change of bases, since similar matrices have identical characteristic polynomials, $\widetilde{\chi}_{U,a}(X)$ does not depend on the particular choice of the basis of U. Moreover, if two modules U and W are isomorphic then $\widetilde{\chi}_{U,a}(X) = \widetilde{\chi}_{W,a}(X)$ for every $a \in \mathcal{A}$. This can be seen at once by choosing bases of U and V, respectively, that are mapped into each other by an isomorphism between U and V. We also note that $\widetilde{\chi}_{U,a}(X)$ is identically one if U is a zero module.

By switching to an appropriate basis for V (starting from a basis of V_1, then extending it to a basis of V_2, and so on), we may assume that the matrices from $\mathcal{A}$ are in block-upper-triangular form. The diagonal blocks correspond to the action specified above on the factors V_i/V_{i-1}. The characteristic polynomial of $a \in \mathcal{A}$ is the product of the characteristic polynomials of the diagonal blocks of a; and the same holds for the variants $\widetilde{\chi}$ we work with. To see this, just observe that both polynomials are determinants of block-upper-triangular matrices (with entries from $F[X]$). The result of our discussion so far can be summarized in the following simple formula:

$$\widetilde{\chi}_a(X) = \widetilde{\chi}_{V,a}(X) = \prod_{i=1}^{t} \widetilde{\chi}_{W_i,a}(X)^{m_i}. \tag{3.2}$$

Let $c \in \mathcal{R}'_j$. The action of c is zero on factors V_i/V_{i-1} which do not belong to ideals J appearing in the definition of $\mathcal{R}'_j$. Thus, if J_i is an ideal not in the sum, then $\widetilde{\chi}_{W_i,c}(X)$ is identically one. Hence from (3.2) we infer

$$\widetilde{\chi}_c(X) = \prod_{p^j|m_i} \widetilde{\chi}_{W_i,c}(X)^{m_i} = \left(\prod_{p^j|m_i} \widetilde{\chi}_{J_i,c}(X)^{m_i/p^j} \right)^{p^j}. \tag{3.3}$$

We denote by $\mathrm{Tr}_{W_i}(a)$ the (ordinary) trace of the action of a on W_i. By comparing the coefficients of X^{p^j} on the two sides, we obtain

$$T_j(c) = \left(\sum_{p^j|m_i} \frac{m_i}{p^j} \mathrm{Tr}_{W_i}(c) \right)^{p^j}. \tag{3.4}$$

From this we see that $T_j(c)$ is the p^jth power of a linear function. It follows that $T_j(c)$ is p^j-semilinear (property 2 of the lemma) on $\mathcal{R}'_j$. Statement 3 follows at once from (3.4) and the identities $\mathrm{Tr}_{W_i}(ab) = \mathrm{Tr}_{W_i}(ba)$ $(a, b \in \mathcal{A})$.

Let $j \leq l$ and $a \in \mathcal{R}'_{j+1}$ be an arbitrary element. Equation (3.3) with $c = a$ and $j+1$ in place of j shows that $\widetilde{\chi}_a(X)$ is the p^{j+1}th power of another polynomial, hence the coefficients of the terms X^t with exponent t not divisible by p^{j+1} are all 0. In particular, $T_j(a) = 0$. This establishes property 4.

It remains to prove statement 5. Assume that $p^j|m_i$ (i.e., $J_i \subseteq \mathcal{R}'_j$) but m_i is not divisible by p^{j+1}. By Lemma 3.3, applied to $\overline{J} = J/\mathrm{Rad}\mathcal{A}$, there exists an element $a \in J$ such that $\mathrm{Tr}_{W_i}(a) = 1$. Let $m = m_i/p^j$. The elements of J act as zero on the composition factors V_h/V_{h-1} which do not belong to J; therefore we have $\widetilde{\chi}_a(X) = \widetilde{\chi}_{W_i,a}(X)^{m_i} = (\widetilde{\chi}_{W_i,a}(X)^m)^{p^j}$. This shows that $T_j(a)$ is the coefficient of X in the polynomial $\widetilde{\chi}_{W_i',a}(X)^m$. We have $T_j(a) = m\mathrm{Tr}_{W_i}(a) = m$. The integer m is not 0 in F. We have finished the proof of the lemma.

The last lemma provides a tool to inductively verify that the subsets $\mathcal{R}_j$ coincide with the ideals $\mathcal{R}'_j$ defined in Lemma 3.4.

Lemma 3.5. *Keeping the notation of Lemma 3.4, for each $j \in \{0, \ldots, l\}$ we have*

$$\mathcal{R}'_{j+1} = \left\{a \in \mathcal{R}'_j \,|\, T_j(ab) = 0 \textit{ for every } b \in \{I\} \cup \mathcal{A}\right\}.$$

Proof. Let $\mathcal{S}_{j+1}$ stand for the right hand side.

From the definition it is obvious that $\mathcal{R}'_{j+1} \leq \mathcal{R}'_j$. Let $a \in \mathcal{R}'_{j+1}$ and $b \in \mathcal{A} \cup \{I\}$. Since $\mathcal{R}'_{j+1}$ is an ideal, we have $ab \in \mathcal{R}'_{j+1}$. Hence by statement 4 of Lemma 3.4, $T_j(ab) = 0$. We obtained $\mathcal{R}'_{j+1} \subseteq \mathcal{S}_{j+1}$.

It remains to prove the reverse inclusion. Based on the semilinearity of T_j on $\mathcal{R}'_j$, we show first that $\mathcal{S}_{j+1}$ is a linear subspace. To this end, let $a, b \in \mathcal{S}_{j+1}$. This means that $a, b \in \mathcal{R}'_j$ and $T_j(ca) = T_j(cb) = 0$ for every $c \in \mathcal{A} \cup \{I\}$. Let $\alpha, \beta \in F$ be arbitrary scalars. We have to show that $\alpha a + \beta b \in \mathcal{S}_{j+1}$. Since $\alpha a + \beta b \in \mathcal{R}'_j$, this is true iff $T_j(c(\alpha a + \beta b)) = 0$ for every $c \in \mathcal{A} \cup \{I\}$. The latter equality follows at once from $T_j(c(\alpha a + \beta b)) = \alpha^{p^j} T_j(ca) + \beta^{p^j} T_j(cb)$.

From the definition and property 3 of Lemma 3.4 it is immediate that $\mathcal{S}_{j+1}$ is closed with respect to multiplication both from the left and from the right by elements from $\mathcal{A}$, hence $\mathcal{S}_{j+1}$ is an ideal of $\mathcal{A}$.

Assume now by way of contradiction, that $\mathcal{S}_{j+1} > \mathcal{R}'_{j+1}$ (strict inclusion). Since $\mathrm{Rad}(\mathcal{A})$ acts as zero on composition factors, we have $\mathcal{S}_{j+1} \supseteq \mathrm{Rad}(\mathcal{A})$.

From these we infer the existence of an index $i \in \{1, \ldots, t\}$ such that $J_i \subseteq \mathcal{S}_{j+1}$, and the multiplicity m_i is divisible by p^j but not by p^{j+1}. By statement 5, Lemma 3.4, there exists an element $a \in J_i$ such that $T_j(a) \neq 0$. This means that $a \notin \mathcal{S}_{j+1}$, a contradiction. We have proved the lemma.

Now we can easily finish the proof of our theorem.

Proof. (of Theorem 3.2) The statements of the theorem are vacuously valid if $\mathcal{A} = \mathrm{Rad}(\mathcal{A})$. We therefore assume that $\mathcal{A} \neq \mathrm{Rad}(\mathcal{A})$. We can use Lemma 3.5 to establish inductively that $\mathcal{R}_j = \mathcal{R}'_j$ $(j = 0, \ldots, l+1)$. Now the sets $\mathcal{R}_j$ are ideals containing the radical by the definition of $\mathcal{R}'_j$. Properties 2 and 3 follow from the statements of Lemma 3.4. This concludes the proof of the theorem.

Remark 3.6. The approach presented here is a simplified and specialized version of a result from [5], where arbitrary fields of characteristic p are allowed. In that general case the characterization of the ideals $\mathcal{R}_j$ is slightly more complicated than formula (3.1). In [19] a radical algorithm over transcendental extensions of finite fields is presented.

The results obtained so far can be summarized as follows.

Theorem 3.7. *Let $\mathcal{A}$ be a finite dimensional algebra over the field $\mathbb{F}_q$ given by a collection of structure constants. Then (a basis of)* $\mathrm{Rad}(\mathcal{A})$ *can be computed in time polynomial in* $\dim_{\mathbb{F}_q} \mathcal{A}$ *and* $\log q$.

Example 3.8. **We explain how the algorithm works for the group algebra $\mathbf{F}_2[\mathbf{S}_3]$ discussed in Example 2.4. We use the regular representation to** embed $\mathbb{F}_2[\mathrm{S}_3]$ into the matrix algebra $\mathrm{M}_6(\mathbb{F}_2)$. We know that $\mathbb{F}_2[\mathrm{S}_3]$ has a one-dimensional radical $\mathcal{I}$ and $\mathbb{F}_2[\mathrm{S}_3]/\mathcal{I} \cong \mathbb{F}_2 \oplus \mathrm{M}_2(\mathbb{F}_2)$. It follows that there are two isomorphism classes of simple modules: a one-dimensional module corresponding to the regular representation of $\mathbb{F}_2$, and a two-dimensional one corresponding to the usual representation of $\mathrm{M}_2(\mathbb{F}_2)$ on ${\mathbb{F}_2}^2$.

Using the structure of $\mathbb{F}_2[\mathrm{S}_3]$, we exhibit an explicit composition series of the regular representation and determine the multiplicities. We denote by W_1 the one-dimensional ideal of $\mathbb{F}_2[\mathrm{S}_3]/\mathcal{I}$. Let W_2 and W_3 stand for the two minimal left ideals of the four-dimensional component consisting of the 2×2 matrices with all zeros in the first and second column, respectively. The factor $\mathbb{F}_2[\mathrm{S}_3]/\mathcal{I}$ is the direct sum of these three minimal left ideals. We denote by ϕ the natural map $\mathbb{F}_2[\mathrm{S}_3] \to \mathbb{F}_2[\mathrm{S}_3]/\mathcal{I}$ and set $V_0 = (0)$, $V_1 = \mathcal{I}$, $V_2 = \phi^{-1}(W_1)$, $V_3 = \phi^{-1}(W_1 + W_2)$, and $V_4 = \mathbb{F}_2[\mathrm{S}_3]$. Obviously, the factors V_2/V_1, V_3/V_2 and V_4/V_3 are simple. It is also straightforward to verify that $\mathbb{F}_2[\mathrm{S}_3]$ does not act as zero on $\mathcal{I}$; therefore the first composition factor V_1/V_0 is a simple module as well. Comparing the dimensions, we obtain that the multiplicity of the one-dimensional simple module is 2, and the same holds for the multiplicity of the 2-dimensional simple module. Hence in Theorem 3.2 we have $\mathcal{R}_1 = \mathbb{F}_2[\mathrm{S}_3]$ and $\mathcal{R}_2 = \mathcal{R}_3 = \mathcal{I}$, i.e., our algorithm finds the radical in the second round.

Exercise 3.9. Let $a \in \mathrm{M}_n(F)$ be a matrix over F, $\operatorname{char} F = 0$. Show that a is nilpotent iff $\mathrm{Tr}(a^i) = 0$ for $i = 1, \ldots, n$. (Hint: Newton's identities for the elementary symmetric polynomials.)

Exercise 3.10. (This is a weak form of a theorem of Wedderburn.) Assume $\operatorname{char} F = 0$, and that $\mathcal{A}$ has an F-basis consisting of nilpotent elements. Prove that every element of $\mathcal{A}$ is nilpotent.

Exercise 3.11. Assume that $\operatorname{char} F = 0$. Show that the sum $a + b$ of two strongly nilpotent elements $a, b \in \mathcal{A}$ is strongly nilpotent again. Use this to prove that $\mathrm{Rad}(\mathcal{A})$ is an ideal of $\mathcal{A}$. (Hint: See the previous exercise.)

Exercise 3.12. Prove Dickson's Theorem. (Hint: Exercise 3.9.)

Exercise 3.13. Show that Dickson's Theorem fails badly in positive characteristic: give an example of a semisimple algebra of matrices over $\mathbb{F}_p$ on which the trace function identically vanishes.

4. Applications to Lie Algebras

We recall first some basic facts about Lie algebras. Detailed expositions can be found in Jacobson [20] and Humphreys [18]. A linear space $\mathcal{L}$ over the field F is a *Lie algebra*, if $\mathcal{L}$ is equipped with an F-bilinear binary operation $[\,]$ such that $[xx] = 0$ for every $x \in \mathcal{L}$ and $[[xy]z] + [[yz]x] + [[zx]y] = 0$ (the Jacobi identity) for every $x, y, z \in \mathcal{L}$.

Just like in the associative case, we have the familiar notions of *subalgebra*, *ideal*, *factor algebra* and *homomorphism* for Lie algebras. The *derived series* of $\mathcal{L}$ is the collection $\mathcal{L}^{(j)}$ of ideals in $\mathcal{L}$ defined as $\mathcal{L}^{(0)} = \mathcal{L}$, and $\mathcal{L}^{(i+1)} = [\mathcal{L}^{(i)}\mathcal{L}^{(i)}]$ for $i > 0$. A Lie algebra $\mathcal{L}$ is called *solvable* if the derived series reaches (0) in finitely many steps: $\mathcal{L}^{(n)} = (0)$ for some natural number n. Here we consider finite dimensional Lie algebras only. In this case $\mathcal{L}$ has a unique maximal solvable ideal, denoted by $R(\mathcal{L})$, the *radical* of $\mathcal{L}$.

The *descending central series* of $\mathcal{L}$ is the sequence $\mathcal{L}^j$ of ideals of $\mathcal{L}$, where $\mathcal{L}^0 = \mathcal{L}$ and $\mathcal{L}^{i+1} = [\mathcal{L}\mathcal{L}^i]$ for $i \geq 0$. A Lie algebra $\mathcal{L}$ is *nilpotent* if $\mathcal{L}^n = (0)$ for some natural number n. If $\dim_F \mathcal{L} < \infty$, then $\mathcal{L}$ has a unique maximal nilpotent ideal $N(\mathcal{L})$, the *nilradical* of $\mathcal{L}$.

Example 4.1. Let $\mathcal{A}$ be an associative algebra over F. For two elements $a, b \in \mathcal{A}$ we write $[ab] = ab - ba$ for the additive commutator. It is easy to check that this operation satisfies the identities of a Lie-bracket. As a consequence, if an F-subspace $\mathcal{L}$ of $\mathcal{A}$ is closed with respect to the operation [ ], then $\mathcal{L}$ can be considered as a Lie algebra. Particularly important are the Lie subalgebras of this form which are obtained from $\mathcal{A} = \mathrm{M}_d(F)$. They are called *linear Lie algebras.*

There is a straightforward analogue of the regular representation for a Lie algebra $\mathcal{L}$. For an $x \in \mathcal{L}$, let $\mathrm{ad}(x) : \mathcal{L} \to \mathcal{L}$ be the linear map that maps $y \in \mathcal{L}$ to $[xy]$. The map $x \mapsto \mathrm{ad}(x)$ is a Lie algebra homomorphism from $\mathcal{L}$ to the linear Lie algebra $\mathrm{gl}(\mathcal{L})$ of all linear transformations of the F-space $\mathcal{L}$. Unfortunately, this map is far from being faithful (if $\mathcal{L}$ is simple, then this map is faithful). We just remark here that, according to a deep theorem of Ado and Iwasawa ([20], Chapter 6), every finite dimensional Lie algebra is actually isomorphic to a linear Lie algebra.

Just like associative algebras, Lie algebras can be conveniently described by structure constants. If $\mathcal{L}$ is a Lie algebra over field F and $a_1, a_2, \ldots, a_n$ is a basis of $\mathcal{L}$, then the bracket is described if we have the products $a_i a_j$ as linear combinations of the basis elements:

$$[a_i a_j] = \gamma_{ij1} a_1 + \cdots + \gamma_{ijn} a_n.$$

The coefficients $\gamma_{ijk} \in F$ are called *structure constants.*

Now we outline algorithms for computing the nilpotent and the solvable radical of a Lie algebra. These problems can be reduced to associative radical computations. First we consider the nilradical. We need a theorem of Jacobson [20] p. 36. Let $\mathcal{L}$ be a finite dimensional Lie algebra over an arbitrary field F.

Jacobson's Theorem 4.2. *Let $\mathcal{A}$ be the associative (matrix-) algebra generated by the linear transformations* $\mathrm{ad}(x)$, $x \in \mathcal{L}$, *i.e., the image* $\mathrm{ad}(\mathcal{L})$ *of the adjoint representation of $\mathcal{L}$. Then an element $x \in \mathcal{L}$ is in the nilradical $N(\mathcal{L})$ if and only if* $\mathrm{ad}(x) \in \mathrm{Rad}(\mathcal{A})$.

This result offers a reasonable way to computing $N(\mathcal{L})$ if the ground field F is a finite field or an algebraic number field. Indeed, we can compute first a basis of $\mathcal{A}$, and then compute $\mathrm{Rad}(\mathcal{A})$ with the algorithms of the previous section. We calculate the intersection of the F-subspaces $\mathrm{ad}(\mathcal{L})$ and $\mathrm{Rad}(\mathcal{A})$ by solving a system of linear equations. By Jacobson's Theorem the inverse image in $\mathcal{L}$ of the intersection $\mathrm{ad}(\mathcal{L}) \cap \mathrm{Rad}(\mathcal{A})$ is $N(\mathcal{L})$. A formal description of our method reads as follows.

```
$Nilradical(\mathcal{L}) :=$
        $\mathcal{A} :=$ associative algebra generated by $\mathrm{ad}(\mathcal{L})$;
        return $\mathrm{ad}^{-1}(\mathrm{Rad}(\mathcal{A}))$.
```

Corollary 4.3. *Let $\mathcal{L}$ be a finite dimensional Lie algebra over the field F, where F is either a finite field or an algebraic number field. Suppose that $\mathcal{L}$ is given as a collection of structure constants. Then the nilradical $N(\mathcal{L})$ can be computed in time polynomial in the input size.*

Next we address the problem of computing the solvable radical $R(\mathcal{L})$. Over fields of characteristic zero, Beck, Kolman and Stewart [2] have given an efficient algorithm to compute $R(\mathcal{L})$. The algorithm hinges on a description of $R(\mathcal{L})$ via the Killing form, which is analogous to Dickson's Theorem. Over fields of positive characteristic, just like Dickson's Theorem, this description is no longer valid.

For finite F the problem of computing $R(\mathcal{L})$ can be reduced efficiently to the problem of computing $N(\mathcal{L})$.

We observe first that $N(\mathcal{L}) \leq R(\mathcal{L})$ and if $N(\mathcal{L}) = (0)$ then $R(\mathcal{L}) = (0)$, because the next to last element of the derived series of $R(\mathcal{L})$ is an Abelian, hence nilpotent ideal of $\mathcal{L}$. With these in mind we define the sequence $\mathcal{L}_i$ of Lie algebras as follows: let $\mathcal{L}_0 = \mathcal{L}$; if $N(\mathcal{L}_i) \neq (0)$ then let $\mathcal{L}_{i+1} = \mathcal{L}_i / N(\mathcal{L}_i)$; if $N(\mathcal{L}_i) = (0)$ then $\mathcal{L}_{i+1}$ is not defined. This sequence of Lie algebras has no more than $\dim_F \mathcal{L} + 1$ elements. From Corollary 4.3 we obtain that the algebras $\mathcal{L}_i$ can all be computed in polynomial time over finite F. Let $\mathcal{L}_j$ be the last algebra of the sequence. We then have $\mathcal{L}_j \cong \mathcal{L}/R(\mathcal{L})$. Moreover, we can construct a basis for $R(\mathcal{L})$ by keeping track of the preimages of the ideals we factored out during the computation of the sequence $\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_j$.

It is instructive to view this argument/computation in terms of ideals of $\mathcal{L}$. For $i > 0$ let $\mathcal{I}_i$ denote the kernel of the composition of the natural maps $\mathcal{L}_0 \to \mathcal{L}_1 \to \cdots \to \mathcal{L}_i$. We then have $\mathcal{I}_1 \subset \mathcal{I}_2 \subset \ldots \subset \mathcal{I}_j$, $N(\mathcal{L}/\mathcal{I}_i) = \mathcal{I}_{i+1}/\mathcal{I}_i$ for $0 < i < j$, and $\mathcal{I}_j = R(\mathcal{L})$. From a basis of $\mathcal{I}_i$ a basis of $\mathcal{I}_{i+1}$ is obtained by a single call of the nilradical-algorithm with the algebra $\mathcal{L}/\mathcal{I}_i$ as input. As a result, we obtain elements $b_1, b_2, \ldots, b_k \in \mathcal{L}$ such that $b_1 + \mathcal{I}_i, \ldots, b_k + \mathcal{I}_i$ form a basis of $\mathcal{I}_{i+1}/\mathcal{I}_i$. Now the elements b_l together with a basis of $\mathcal{I}_i$ will constitute a basis of $\mathcal{I}_{i+1}$. In j such rounds we obtain a basis of $R(\mathcal{L})$. Below we give a formal description of the algorithm.

```
$SolvableRadical(\mathcal{L}) :=$
        $S := (0)$;
        loop
          $\overline{S} := Nilradical(\mathcal{L}/S)$;
          $\phi :=$ natural map $\mathcal{L} \to \mathcal{L}/S$;
          $S := \phi^{-1}(\overline{S})$;
        until $\overline{S} = (0)$;
        return $S$.
```

Corollary 4.4. *Let $\mathcal{L}$ be a finite dimensional Lie algebra over $\mathbb{F}_q$, given by structure constants. Then (a basis of) the solvable radical $R(\mathcal{L})$ can be computed in time polynomial in* $\dim_{\mathbb{F}_q}\mathcal{L}$ *and* $\log q$.

Variants of the radical algorithms discussed here are implemented by Willem de Graaf in a general library of Lie algebra algorithms, called ELIAS (for Eindhoven LIe Algebra System), which is built into the computer algebra systems GAP4 and MAGMA. These activities are part of a bigger project, called ACELA, which aims at an interactive book on Lie algebras (cf. [6]).

5. Finding the Simple Components of Semisimple Algebras

We have given an algorithm for computing $\text{Rad}(\mathcal{A})$. Our next target is the Wedderburn decomposition of the radical-free part $\mathcal{A}/\text{Rad}(\mathcal{A})$. From $\mathcal{A}$ we can form $\mathcal{A}/\text{Rad}(\mathcal{A})$ efficiently. We therefore can assume that $\mathcal{A}$ is semisimple, that is $\text{Rad}(\mathcal{A}) = (0)$. The idea of the algorithm is easier to explain when the ground field F is finite; we will consider this problem first. Then the necessary modifications to make the method work over $\mathbb{Q}$ will be outlined. The input to the problem is a finite semisimple associative algebra $\mathcal{A}$ over the field $\mathbb{F}_q$ ($q = p^s$, p prime). $\mathcal{A}$ is given as an array of structure constants. By Wedderburn's Theorem there exists a decomposition

$$\mathcal{A} = \mathcal{A}_1 \oplus \mathcal{A}_2 \oplus \cdots \oplus \mathcal{A}_k,$$

where $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ are the minimal nonzero ideals of $\mathcal{A}$. We give an f-algorithm running in time polynomial in $\dim_{\mathbb{F}_q}\mathcal{A}$ and $\log q$ to find bases for the ideals $\mathcal{A}_i$. This method was given by K. Friedl [13].

First we reduce the problem to the case where $\mathcal{A}$ is commutative. Recall that the centre $C(\mathcal{A})$ of $\mathcal{A}$ is

$$C(\mathcal{A}) = \{x \in \mathcal{A} \mid xy = yx \text{ for every } y \in \mathcal{A}\}.$$

It is not difficult to show that $C(\mathcal{A})$ is also a semisimple algebra. Moreover, the Wedderburn decomposition of $C(\mathcal{A})$ is inherited from the Wedderburn decomposition of $\mathcal{A}$ in the following sense:

$$C(\mathcal{A}) = C(\mathcal{A}_1) \oplus C(\mathcal{A}_2) \oplus \cdots \oplus C(\mathcal{A}_k).$$

Let $a_1, a_2, \ldots, a_n$ be the input basis of $\mathcal{A}$, i.e., the one with respect to which the structure constants are given. From $a_1, a_2, \ldots, a_n$ we can obtain a basis of $C(\mathcal{A})$ simply by solving a system of linear equations. The fact that an element $z \in \mathcal{A}$ is in $C(\mathcal{A})$ is equivalent to the relations $za_i = a_i z$, $i = 1, \ldots, n$. If we write z as a linear combination with 'unknown' coefficients $z = \lambda_1 a_1 + \lambda_2 a_2 + \cdots + \lambda_n a_n$, then we obtain a system of linear equations over $\mathbb{F}_q$. A basis of the solution-space gives a basis of $C(\mathcal{A})$.

From the Wedderburn decomposition of $C(\mathcal{A})$ we can easily recover the Wedderburn decomposition of $\mathcal{A}$. In fact, we have

$$\mathcal{A}_i = C(\mathcal{A}_i)\mathcal{A} \quad \text{for} \quad i = 1, \ldots, k. \tag{5.1}$$

Indeed, $C(\mathcal{A}_i)\mathcal{A}$ is a nonzero ideal of $\mathcal{A}$ and clearly $C(\mathcal{A}_i)\mathcal{A} \leq \mathcal{A}_i$. Now (5.1) follows from the simplicity of $\mathcal{A}_i$. The transition from $C(\mathcal{A}_i)$ to $\mathcal{A}_i$ is simple linear algebra. We select a maximal linearly independent set from the products $b_j a_r$, where b_j and a_r run through a basis of $C(\mathcal{A}_i)$ and $\mathcal{A}$, respectively.

Henceforth we assume that $\mathcal{A}$ is a commutative semisimple algebra over $\mathbb{F}_q$. In this case the ideals $\mathcal{A}_i$ are finite fields containing the ground field $\mathbb{F}_q$. We describe a method which breaks $\mathcal{A}$ into a direct sum of smaller ideals $\mathcal{A} = \mathcal{I} \oplus \mathcal{J}$, unless $k = 1$. This cutting procedure suffices to solve the problem, because $\mathcal{I}$ and $\mathcal{J}$ are also semisimple and commutative. Moreover, their ideals are also ideals of $\mathcal{A}$. We can therefore cut $\mathcal{I}$ and $\mathcal{J}$, and so on, until we have all the minimal ideals $\mathcal{A}_i$.

The procedure for cutting $\mathcal{A}$ is an iteration which processes sequentially the basis $a_1, \ldots, a_n$. As for the general step of the iteration, suppose that the elements $a_1, \ldots, a_i$ have already been processed and that the subalgebra F_i generated by $a_1, \ldots, a_i$ is a field. This is certainly true at the beginning with $F_0 = \mathbb{F}_q$. If $i = n$, then by assumption $F_n = \mathcal{A}$ is a field; therefore it has no proper ideals. Suppose that $i < n$. We calculate first the (monic) minimal polynomial $f(X)$ of the element $b = a_{i+1}$ over the field F_i. This is again easy linear algebra. We have to find the first linear dependence over F_i of the elements $1_{\mathcal{A}}, b, b^2, \ldots, b^n$. Next we factor f into irreducible factors over the field F_i. This is the point where we call a factoring oracle. If f turns out to be irreducible over F_i, then $F_i(a_{i+1})$ is a field. In this case we put $F_{i+1} = F_i(a_{i+1})$ and the i-th step is finished. We maintained the loop-invariant because F_{i+1} is a field.

What happens if f is reducible over F_i, say $f = gh$ where $g, h \in F_i[X]$ are nonconstant polynomials? Using the minimality of f and the fact that $\mathcal{A}$ is a direct sum of fields, we see that g and h are relatively prime polynomials:

there exist $g', h' \in F_i[X]$ such that $g'g + h'h = 1$. We set $\mathcal{I} = \mathcal{A}g(b)$ and $\mathcal{J} = \mathcal{A}h(b)$. We have that $\mathcal{I}$ and $\mathcal{J}$ are proper ideals of $\mathcal{A}$, because $\mathcal{I}h(b) = (0)$ and $\mathcal{J}g(b) = (0)$, while $\mathcal{A}h(b) \neq (0)$ and $\mathcal{A}g(b) \neq (0)$. From these facts we see also that $\mathcal{I} \cap \mathcal{J} = (0)$. From the relation $g'(b)g(b) + h'(b)h(b) = 1_{\mathcal{A}}$ we infer that the ideal generated by $\mathcal{I}$ and $\mathcal{J}$ is $\mathcal{A}$. This implies that $\mathcal{A} = \mathcal{I} \oplus \mathcal{J}$. We formalize the cutting method in the following procedure.

```
$Cut(\mathcal{A}) :=$
        $(a_1, a_2, \ldots, a_n) := basis\ of\ \mathcal{A}$;
        $F := \mathbb{F}_q$;
        for $i$ from 1 to $n$ do
          $f(X) := minimal\ polynomial\ of\ a_i\ over\ F$;
          $H := irreducible\ factors\ of\ f(X)\ over\ F$;
          if $|H| > 1$ then
            choose $g(X) \in H$;
            $h(X) := f(X)/g(X)$;
            $\mathcal{I} := g(a_i)\mathcal{A}$;
            $\mathcal{J} := h(a_i)\mathcal{A}$;
            return $\{\mathcal{I}, \mathcal{J}\}$
          fi;
          $F := F(a_i)$
        od;
        return $\{\mathcal{A}\}$.
```

The discussion above implies the following:

Theorem 5.1. *Let $\mathcal{A}$ be a finite semisimple associative algebra over the field $\mathbb{F}_q$ ($q = p^s$, p prime), given by a collection of structure constants. Then there exists an f-algorithm running in time polynomial in* $\dim_{\mathbb{F}_q} \mathcal{A}$ *and* $\log q$ *for finding the Wedderburn decomposition of $\mathcal{A}$.*

We can substitute into the cutting procedure either a polynomial time Las Vegas method [4] or the deterministic method [3] for the oracle for factoring polynomials:

Corollary 5.2. *The minimal ideals of the algebra $\mathcal{A}$ above can be found by a polynomial time Las Vegas algorithm. Similarly, the minimal ideals of $\mathcal{A}$ can be found by a deterministic method running in time polynomial in* $\dim_{\mathbb{F}_q} \mathcal{A}$ *and q.*

Remark 5.3. The cutting procedure works over algebraic number fields as well. There is only one additional difficulty to cope with. During the iterative application of the cutting method the sizes of the intermediate results (bases for ideals) may grow too fast. It is possible however, to fix this problem. One can establish a bound similar to the Landau-Mignotte Theorem in Chapter 4 for the smallest bases of the ideals of $\mathcal{A}$. From an arbitrary basis of an ideal $\mathcal{I}$ it is possible to obtain efficiently a small basis for $\mathcal{I}$. Thus, an application of the cutting procedure is followed by a 'reduction step', in which we calculate small bases of the new ideals found in the cutting phase. This leads to a deterministic polynomial time algorithm for finding the Wedderburn decomposition over algebraic number fields [13].

Remark 5.4. Eberly [10] proposed an efficient Las Vegas algorithm which avoids iteration over the basis elements of $\mathcal{A}$. The key idea is the use of splitting elements. An element $b \in \mathcal{A}$ is a *splitting element* of $\mathcal{A}$ if the minimal polynomial of b over F is squarefree and has maximal degree among the elements of $\mathcal{A}$. If F is a sufficiently large perfect field, and $\mathcal{A}$ is a semisimple commutative algebra over F, then $\mathcal{A}$ contains a splitting element with minimal polynomial of degree $\dim_F \mathcal{A}$. In this case a random element $b \in \mathcal{A}$ has a good chance of being a splitting element.

Let b be a splitting element of $\mathcal{A}$, and let f be the minimal polynomial of b over F with factorization $f = f_1 f_2 \cdots f_k$ into irreducible polynomials $f_i \in F[X]$. It can be shown that $\mathcal{A}f_i(b)$, $i = 1, 2, \ldots, k$ are the minimal ideals of $\mathcal{A}$.

The procedure in GAP4 (and in MAGMA) for computing the Wedderburn decomposition has been implemented by Willem de Graaf. It employs a combination of the ideas described here: the cutting procedure for small F, and splitting elements for large fields.

Exercise 5.5. Suppose that $\mathcal{A}$ is semisimple and commutative. Let $0 \neq a$ be a zero divisor in $\mathcal{A}$. Show that $\mathcal{A}a$ is a proper ideal of $\mathcal{A}$.

Exercise 5.6. (Fast exponentiation) Let $a \in \mathcal{A}$. Show that a^m can be computed using only $O(\log m)$ multiplications in $\mathcal{A}$. (Hint: Consider first the binary representation of m.)

The next four exercises sketch a possible refinement of the method of this section for computing the Wedderburn decomposition of a semisimple commutative algebra over $\mathbb{F}_p$. The method is essentially a reduction to the case where the minimal ideals of $\mathcal{A}$ are all isomorphic to $\mathbb{F}_p$. This implies for example, that the factoring oracle is employed only for finding roots of polynomials in the prime field $\mathbb{F}_p$.

Exercise 5.7. Suppose that $F = \mathbb{F}_p$, where p is a prime and that $a^p = a$ holds for every $a \in \mathcal{A}$. Show that $\mathcal{A}$ is semisimple. Moreover $\mathcal{A}$ is a direct sum of ideals isomorphic to $\mathbb{F}_p$.

Exercise 5.8 (Berlekamp Subalgebra). Suppose that $F = \mathbb{F}_p$. Put

$$B(\mathcal{A}) = \{a \in \mathcal{A} \mid a^p = a\}.$$

Show that $B(\mathcal{A})$ is a subalgebra of $\mathcal{A}$.

Exercise 5.9. Show that a basis of $B(\mathcal{A})$ can be computed using $O(n^3 \log p)$ arithmetical operations over $\mathbb{F}_p$. (Hint: Observe that the map $a \mapsto a^p$ is $\mathbb{F}_p$-linear. Moreover $B(\mathcal{A})$ is precisely the set of fixed points of this map.)

Exercise 5.10. Suppose that $F = \mathbb{F}_p$, and $\mathcal{A}$ is semisimple commutative. Let I be an ideal of $\mathcal{A}$. Show that $D = B(\mathcal{A}) \cap I$ is a nonzero ideal of $B(\mathcal{A})$ and $D\mathcal{A} = I$. (Hint: Wedderburn's Theorem.)

6. Zero Divisors in Finite Algebras

One way to look at the cutting procedure is to note that it finds zero divisors in a finite algebra $\mathcal{A}$. Indeed, if $\mathcal{A}$ is decomposable at all, then we construct a decomposition with the pair of zero divisors $g(b)$ and $h(b)$. We can also find zero divisors efficiently if $\text{Rad}(\mathcal{A}) \neq (0)$: an arbitrary nonzero $x \in \text{Rad}(\mathcal{A})$ and a power of x will suffice. Thus, with the methods discussed so far, we can find zero divisors in (Las Vegas) polynomial time, unless $\mathcal{A}$ is a simple algebra. To cover this remaining case, we study zero divisors of finite simple algebras in more detail.

The argument presented below is a constructive version of the proof in [16] pp. 71–72 of Wedderburn's Theorem on finite division algebras. Let $F = \mathbb{F}_q$ be a finite field, and $a \in \text{M}_n(F)$, $a \notin F$ such that $L = F(a)$ is a field. Let l denote the degree of L over F. Thus, the minimal polynomial f of a over F is irreducible over F and $\deg(f) = l$.

Lemma 6.1.
(i) Let a and F be as above. Then there exists a matrix $c \in \text{M}_n(F)$ such that $c^{-1}ac = a^q$.
(ii) Let $c \in \text{M}_n(F)$ be an element satisfying (i), and denote by $\text{Alg}(a,c)$ *the F-algebra generated by a and c. Then* $\text{Alg}(a,c)$ *is not commutative, and*

$$\text{Alg}(a,c) = L + cL + \cdots + c^m L + \cdots .$$

Proof. By assumption L is a simple subalgebra of $\text{M}_n(F)$ and the automorphism ϕ of L sending a to a^q leaves F element-wise fixed. A theorem of Noether and Skolem (Pierce, [26], Section 12.6) implies that this automorphism is inner; there exists a matrix $c \in \text{M}_n(F)$ such that $b^q = \phi(b) = c^{-1}bc$ for every $b \in L$. Now (i) follows. As for the statements of (ii), $\text{Alg}(a,c)$ is not commutative, because $ac \neq ca$. The rest follows from the obvious relation $Lc \subseteq cL$.

The next statement paves the way for a considerable simplification in the search for zero divisors. Let c be an element satisfying (i) above.

Lemma 6.2. *Suppose that* $\mathrm{Alg}(a,c) = \mathrm{M}_n(F)$. *Then we have* $l = n$, $\dim_F F(c) = n$, $c^n \in F$, *and* $\mathrm{Alg}(a,c)$ *is a direct sum of* F*-subspaces:*

$$\mathrm{Alg}(a,c) = L \oplus cL \oplus \cdots \oplus c^{n-1}L. \tag{6.1}$$

Proof. It is a simple calculation to verify that for an arbitrary natural number i we have $c^{-i}ac^i = a^{q^i}$. Note also, that $a^{q^l} = a$ because the field $F(a)$ has q^l elements. Thus, we have $c^{-l}ac^l = a^{q^l} = a$, hence $ac^l = c^l a$. We infer that c^l is in the centre of $\mathrm{Alg}(a,c) = \mathrm{M}_n(F)$, which is F. Now by Lemma 6.1 $c^l \in F \subset L$ implies that

$$\mathrm{Alg}(a,c) = L + cL + \cdots + c^{l-1}L.$$

From this we infer that $\dim_F \mathrm{Alg}(a,c) \leq l^2$. Moreover, there is equality here if and only if the sum is a direct sum. Recall that l is the degree of the minimal polynomial of a, hence $l \leq n$. In these circumstances $\mathrm{Alg}(a,c) = \mathrm{M}_n(F)$ is possible only if $l = n$, and the sum in Lemma 6.1 (ii) is a direct sum. Finally we observe that c cannot be a root of a polynomial over F with degree $k < n$. Otherwise $\mathrm{Alg}(a,c)$ would be the direct sum of the subspaces $c^j L$ $(0 \leq j < k)$. By counting dimensions we see that such a short decomposition is impossible.

Our algorithmic task will eventually boil down to finding zero divisors in an algebra of the form $\mathrm{M}_k(F)$, described by structure constants with respect to a 'general' basis. The point is that here we do not have a nice basis of the algebra (say of rank one matrices). For this reason we have to develop a basis-free approach to the problem.

We study further the setting at hand: we have $a, c \in \mathrm{M}_n(F)$ such that $L = F(a)$ is a field, $c^{-1}ac = a^q$, and $\mathrm{Alg}(a,c) = \mathrm{M}_n(F)$. We have established that c is a root of a polynomial of the form $X^n - \alpha$ where $\alpha \in F$, and this is the minimal polynomial of c over F. Recall that the *norm* of an element d of L is defined as

$$\mathrm{N}(d) := dd^q d^{q^2} \cdots d^{q^{n-1}}.$$

(More precisely this is the norm of L over F.) The following lemma reduces the problem of zero divisors to finding elements with a specified norm.

Lemma 6.3. *Let* $d \in L$ *be an element such that* $\mathrm{N}(d) = 1/\alpha$. *Then* $1 - cd$ *is a zero divisor in* $\mathrm{Alg}(a,c) = \mathrm{M}_n(F)$.

Proof. Consider the element $z \in \mathrm{Alg}(a,c)$ defined by the following expression

$$z = 1 + cd + c^2 dd^q + \cdots + c^{n-1} dd^q \cdots d^{q^{n-2}}.$$

It is a direct calculation to verify that $z(1 - cd) = 0$. On the other hand, the decomposition in (6.1) is a direct sum. This implies that neither z nor $1 - cd$ can be zero, as both of them have 1 for the component belonging to L. This completes the proof.

Unfortunately we cannot treat the norm equation $\mathrm{N}(X) = 1/\alpha$ (with X as unknown) directly as an instance of the factoring problem, because of the high degree of the polynomial involved. We circumvent this difficulty by imposing additional conditions on c.

Lemma 6.4. *Let a, c be as before. Suppose further that the minimal polynomial $X^n - \alpha$ of c is irreducible over F. Then the polynomial $g(X) = X^n - 1/\alpha$ is also irreducible over F. Moreover, g splits into linear factors in L, and if n is odd then $\mathrm{N}(d) = 1/\alpha$ for each $d \in L$ with $g(d) = 0$.*

Proof. From the irreducibility of $X^n - \alpha$ we infer that $F(c)$ is a field and $\dim_F F(c) = n$. It is clear also, that c and $1/c$ generate the same field over F, hence the minimal polynomial of $1/c$ is also irreducible over F with degree n. We have $g(X) \in F[X]$, $g(1/c) = 0$ and $\deg(g) = n$. From this we infer that g is the minimal polynomial of $1/c$ over F, hence g is irreducible over F. The fields L and $F(c)$ have the same degree over F, hence $F(c) \cong L$, so g splits into linear factors in L. Finally, let d be a root of g from L. The irreducibility of g implies that $g(X) = (X - d)(X - d^q) \cdots (X - d^{q^{n-1}})$; therefore the constant term is $(-1)^n \mathrm{N}(d) = -\mathrm{N}(d) = -1/\alpha$, proving the last claim.

Lemma 6.4 shows that if n is odd, then we can solve the norm equation $\mathrm{N}(X) = 1/\alpha$ by factoring a polynomial of degree n. The next statement gives a similar result for $n = 2$.

Lemma 6.5. *Let L be a quadratic extension field of F. Suppose that we are given an element $\beta \in F$ such that $X^2 - \beta$ is an irreducible polynomial over F (in other words, β is a quadratic nonresidue in F). Then we can find an element $d \in L$ such that $\mathrm{N}(d) = \beta$ by a polynomial time f-algorithm.*

Proof. Suppose first that $-\beta$ is a quadratic nonresidue in F. If $d \in L$ is a root of $h(X) = X^2 + \beta$ then the other root of h is d^q. By inspecting the constant term of $h(X) = (X - d)(X - d^q)$ we obtain that $d^{q+1} = \mathrm{N}(d) = \beta$. Thus, a required d can be found by factoring a quadratic polynomial in L.

It remains to consider the case where $-\beta$ is a quadratic residue in F. Let γ be an element of F such that $\gamma^2 = -\beta$. Such γ can be found by factoring a quadratic polynomial in F. We claim that it suffices to find an element $b \in L$ such that $\mathrm{N}(b) = -1$. Indeed, then we can put $d = \gamma b$, because $\mathrm{N}(d) = \gamma\gamma^q \mathrm{N}(b) = \gamma^2 \mathrm{N}(b) = (-\beta) \cdot (-1) = \beta$.

We now turn to the norm equation $\mathrm{N}(X) = -1$. If $h \in L$ is a quadratic nonresidue in L, then by Euler's lemma we have

$$h^{\frac{q^2-1}{2}} = h^{\frac{(q-1)(q+1)}{2}} = -1.$$

Thus, for $b = h^{\frac{q-1}{2}}$ we have

$$\mathrm{N}(b) = bb^q = h^{\frac{q-1}{2}} h^{\frac{q-1}{2}q} = h^{\frac{(q-1)(q+1)}{2}} = -1.$$

To find a good h, we define a sequence $z_1, z_2, \ldots, z_k$ of elements of L: let $z_1 = -1$. Suppose that z_i is defined. Then let z_{i+1} be an element of L such that ${z_{i+1}}^2 = z_i$, provided that such an element exists. Let z_k be the last element of this sequence. We have

$$z_k^{2^{k-1}} = -1 \text{ and } z_k^{2^k} = 1,$$

hence z_k generates a multiplicative subgroup of order 2^k in L. This implies in particular, that $2^k \leq q^2$, hence z_k is obtained at the expense of solving at most $2\log_2 q$ quadratic equations in L. Also, $h = z_k$ is a good choice, because z_k is not a quadratic residue in L.

In both cases the norm equation can be solved by factoring not too many quadratic polynomials. The proof is complete.

We can now describe a polynomial time f-algorithm for finding zero divisors in finite algebras. The input is an algebra $\mathcal{A}$ over the finite field $Z = \mathbb{F}_{p^r}$, p prime and $\dim_Z \mathcal{A} = m$. The algebra $\mathcal{A}$ is specified as a collection of structure constants. The procedure ZERODIV() returns a pair of zero divisors $x, y \in \mathcal{A}$, if zero divisors in $\mathcal{A}$ exist.

procedure ZERODIV($\mathcal{A}$)

Step 1. *Compute first* Rad($\mathcal{A}$) *with the method of Section 3. If* Rad($\mathcal{A}$) $\neq$ (0) *then let* $x \in$ Rad($\mathcal{A}$) *be an arbitrary nonzero element. As* x *is nilpotent, an appropriate power of* x *may serve as* y; **return**(x, y).

Step 2. ($\mathcal{A}$ is semisimple)
With the method of Section 5 compute the Wedderburn decomposition of $\mathcal{A}$. *If* $\mathcal{A}$ *is not simple, with* $\mathcal{I}$ *and* $\mathcal{J}$ *different minimal ideals of* $\mathcal{A}$, *then let* x *and* y *be arbitrary nonzero elements of* $\mathcal{I}$ *and* $\mathcal{J}$, *respectively;* **return**(x, y).

Step 3. ($\mathcal{A}$ is simple)
Check whether $\mathcal{A}$ *is commutative. This involves comparisons of the products* $a_i a_j$ *and* $a_j a_i$, *where* $a_1, \ldots, a_m$ *is a basis of* $\mathcal{A}$. *If* $\mathcal{A}$ *is commutative, then terminate concluding that* $\mathcal{A}$ *is a field (and hence it has no zero divisors).*

Step 4. ($\mathcal{A}$ is a full matrix algebra over an extension field $F = \mathbb{F}_q$ of Z; say $\mathcal{A} \cong \mathrm{M}_n(F)$ and $n > 1$.)
Select an element $b \in \mathcal{A}$ *which is not in* $F \cdot 1_{\mathcal{A}}$. *Compute and factor the minimal polynomial* $f(X)$ *of* b *over* F. *If* f *is not irreducible over* F, *say* $f(X) = g(X)h(X)$ *is a proper factorization, then* **return**($g(b), h(b)$).

Step 5. (Here f is irreducible over F, hence $F(b)$ is a field.)
If $\dim_F F(b)$ ***is even, then select an*** $a \in F(b)$ ***such that*** $\dim_F F(a) = 2$ ***(find a solution of the system of*** F***-linear equations*** $z^{q^2} = z$ ***in*** $F(b) \setminus F$***). If*** $\dim_F F(b)$ ***is odd, then put*** $a := b$.

Step 6. ($F(a)$ is a field and $l = \dim_F F(a)$ is either odd, or $l = 2$.)
By solving a system of linear equations over F find a nonzero $c \in \mathcal{A}$ such that $ac = ca^q$. Compute and factor the minimal polynomial $g(X)$ of c over F. If g is not irreducible over F, then **return** *zero divisors as in Step 4.*

Step 7. (The element $c \in \mathcal{A}$ is invertible and $c^{-1}ac = a^q$.)
Form (a basis of) $\mathrm{Alg}(a, c)$, *the F-algebra generated by a and c. If* $\mathrm{Alg}(a, c) \neq \mathcal{A}$ *then set* $\mathcal{A} := \mathrm{Alg}(a, c)$ *and* **go back to** *Step 1.*

Step 8. (Here $\mathrm{Alg}(a, c) = \mathcal{A} \cong \mathrm{M}_n(F)$ with $n = \sqrt{\dim_F \mathcal{A}}$. Moreover n is either odd or $n = 2$. The minimal polynomial of c over F is $f(X) = X^n - \alpha$ for some $\alpha \in F$, and f is irreducible over F.)
Find a solution $d \in F(a)$ of the norm equation $\mathrm{N}(X) = 1/\alpha$. *Use the method of Lemma 6.4 for this if n is odd, and the algorithm of Lemma 6.5 for $n = 2$. Finally set $v := 1 - cd$, and*

$$u := 1 + cd + c^2 dd^q + \cdots + c^{n-1} dd^q \cdots d^{q^{n-2}}.$$

return(u, v).

end procedure

Theorem 6.6. *Let $\mathcal{A}$ be an algebra over $\mathbb{F}_{p^r}$ given as a collection of structure constants,* $\dim_{\mathbb{F}_{p^r}} \mathcal{A} = m$. *Then* ZERODIV() *finds a pair of zero divisors in $\mathcal{A}$, unless $\mathcal{A}$ is a field (and hence contains no zero divisors). As an f-algorithm* ZERODIV() *runs in time polynomial in the input size. In other words, the running time is bounded by a polynomial in m, r, and* $\log p$.

Proof. First we consider the correctness of the method. If $\mathcal{A}$ is not simple, then we clearly find zero divisors at Step 1 or 2. If $\mathcal{A}$ is a field then ZERODIV terminates at Step 3 with a correct answer again. From Step 4 on we are in a full matrix algebra and work towards the situation described in the annotation of Step 8. At Steps 4–6 we either find a and c with the required properties, or obtain zero divisors during the process.

If $\mathrm{Alg}(a, c) \neq \mathcal{A}$ then we reduce the problem to a smaller instance. The new $\mathcal{A}$ is not commutative by Lemma 6.1, hence also contains zero divisors. The dimension of $\mathcal{A}$ decreases here. As a consequence, a Step of ZERODIV is executed at most m times. Lemma 6.2 implies that the annotation preceding Step 8 is valid, hence Lemma 6.3 is applicable: u and v is indeed a pair of zero divisors.

As for the complexity of the method, it suffices to establish a polynomial bound on the time required by a single Step, because Step 7 allows for iteration at most m times. For Steps 1 and 2 polynomial bounds follow from Theorems 3.7 and 5.1. Step 3 is carried out by inspecting and comparing $2m(m-1)$ products of shape $a_i a_j$. Steps 4–7 require linear algebra and factoring polynomials of degree at most m over F. Finally the norm equation is solved by one of the polynomial time f-algorithms given in Lemmas 6.4 and 6.5.

Remark 6.7. The jump back at Step 7 is not really necessary. It can be shown that at that point $\mathrm{Alg}(a,c) \cong \mathrm{M}_l(F)$. Therefore we could directly proceed with Step 8, with l in the place of n. We leave the details of this to the reader.

As we did with the method for the Wedderburn decomposition, we can substitute into ZERODIV either a polynomial time Las Vegas method [4] or the deterministic method [3] for the oracle for factoring polynomials:

Corollary 6.8. *Let $\mathcal{A}$, m, r, and p be as before. We can find zero divisors in $\mathcal{A}$ (if there are any) in Las Vegas time polynomial in m, r, and* $\log p$. *Also, the problem can be solved in deterministic time polynomial in m, r, and p.*

The results of Sections 3 and 5 imply that there exists an efficient f-algorithm to decide if a given finite algebra $\mathcal{A}$ is isomorphic to a full matrix algebra. One checks if $\mathrm{Rad}(\mathcal{A}) = (0)$ and whether $\mathcal{A}$ is directly indecomposable. If $\mathcal{A}$ turns out to be simple, say if $\mathcal{A} \cong \mathrm{M}_n(\mathbb{F}_q)$, then we can also find n and $\mathbb{F}_q$ efficiently. In Exercises 6.9–6.12 we consider the algorithmic problem to establish an explicit isomorphism from $\mathcal{A}$ to $\mathrm{M}_n(\mathbb{F}_q)$. This means representing $\mathcal{A}$ as an algebra of linear transformations of an n-dimensional $\mathbb{F}_q$-space V.

Exercise 6.9. Let $e \in \mathrm{M}_n(\mathbb{F}_q)$ be an idempotent matrix of rank 1 and put $V = \mathrm{M}_n(\mathbb{F}_q)e$. Show that $\dim_{\mathbb{F}_q} V = n$ and $\mathrm{M}_n(\mathbb{F}_q)$ acts nontrivially, and hence faithfully, on V via multiplication from the left.

Exercise 6.10. Let $e \in \mathrm{M}_n(\mathbb{F}_q)$ be an idempotent such that $rank(e) = m$. Then $e\mathrm{M}_n(\mathbb{F}_q)e \cong \mathrm{M}_m(\mathbb{F}_q)$.

Exercise 6.11. Let $x \in \mathrm{M}_n(\mathbb{F}_q)$ be a singular matrix, and let e be a right identity element of the left ideal $\mathrm{M}_n(\mathbb{F}_q)x$. Prove that e is a singular idempotent.

Exercise 6.12. Suppose that we are given an algebra $\mathcal{A}$ such that $\mathcal{A} \cong \mathrm{M}_n(\mathbb{F}_q)$. Then an explicit isomorphism from $\mathcal{A}$ to $\mathrm{M}_n(\mathbb{F}_q)$ can be constructed by a polynomial time f-algorithm. (Hint: Exercise 6.9 shows that it suffices to find an idempotent e of rank 1. To achieve this, it is enough to give an algorithm to construct a singular idempotent; Exercise 6.10 allows then to reduce the problem to a smaller (in n) instance. To find a singular idempotent, call ZERODIV($\mathcal{A}$). If a zero divisor (a singular matrix) is returned, then use the idea of Exercise 6.11.)

An explicit isomorphism is useful because the usual representation of $\mathrm{M}_n(\mathbb{F}_q)$, as the algebra of all $n \times n$ matrices over $\mathbb{F}_q$, is easy to handle. For example we can conveniently decompose $\mathrm{M}_n(\mathbb{F}_q)$ into a direct sum of minimal left ideals.

Exercise 6.13. Let $e_{ii} \in \mathrm{M}_n(\mathbb{F}_q)$ be the matrix having 1 in position (i,i) and 0 elsewhere. Show that $\mathrm{M}_n(\mathbb{F}_q)e_{ii}$ is a minimal nonzero left ideal of $\mathrm{M}_n(\mathbb{F}_q)$. Moreover,

$$\mathrm{M}_n(\mathbb{F}_q) = \mathrm{M}_n(\mathbb{F}_q)e_{11} \oplus \mathrm{M}_n(\mathbb{F}_q)e_{22} \oplus \cdots \oplus \mathrm{M}_n(\mathbb{F}_q)e_{nn}.$$

Exercise 6.14. Let $\mathcal{A}$ be a given finite semisimple algebra over $\mathbb{F}_q$. Suggest a polynomial time f-algorithm to express $\mathcal{A}$ as a direct sum of minimal left ideals. (Hint: With the method of Section 5 break $\mathcal{A}$ into a direct sum of minimal ideals. The minimal ideals are simple algebras over finite fields. Now use the results of Exercises 6.12–6.13.)

In the subsequent exercises we outline an algorithm to find a nontrivial common invariant subspace for a set $X_1, \ldots, X_k \in \mathrm{M}_n(\mathbb{F}_q)$ of matrices acting on the $\mathbb{F}_q$-space V of column vectors of length n.

Exercise 6.15. Let $\mathcal{A}$ be a subalgebra of $\mathrm{M}_n(\mathbb{F}_q)$ and suppose $\mathrm{Rad}(\mathcal{A}) \neq (0)$. Prove that $\mathrm{Rad}(\mathcal{A})V$ is a proper $\mathcal{A}$-invariant subspace of V.

Exercise 6.16. Let $\mathcal{A}$ be a semisimple subalgebra of $\mathrm{M}_n(\mathbb{F}_q)$ such that $I \in \mathcal{A}$. Consider the decomposition of $\mathcal{A}$ into a direct sum of minimal left ideals: $\mathcal{A} = \rho_1 \oplus \cdots \oplus \rho_m$. Let $0 \neq v$ be an arbitrary vector from V. Prove that there is a j such that $\rho_j v \neq (0)$, and for any such j the subspace $\rho_j v$ is a minimal nonzero $\mathcal{A}$-invariant subspace.

Exercise 6.17. Let $S = \{X_1, \ldots, X_k\} \subset \mathrm{M}_n(\mathbb{F}_q)$ be a set of given matrices. Suggest a polynomial time f-algorithm to find a nontrivial S-invariant subspace of V if there is any. An $\mathbb{F}_q$-subspace $(0) \subset U \subset V$ is required for which $XU \subseteq U$ for every $X \in S$. (Hint: Let $\mathcal{A}$ be the matrix-algebra generated by S. Clearly S and $\mathcal{A}$ have the same invariant subspaces. If $\mathcal{A}V \subset V$, then $U = \mathcal{A}V$ suffices. Otherwise if $\mathrm{Rad}(\mathcal{A}) \neq (0)$, then use Exercise 6.15. For the case $\mathrm{Rad}(\mathcal{A}) = (0)$, see Exercise 6.16.)

Notes

The general algebraic background required to follow the chapter is usually covered in a first course on algebra. We use extensively the basics of linear algebra and the first facts about field extensions. Lang [23] is an excellent text on basic algebra. Mignotte [25] covers the subject with special emphasis on tools, methods relevant in computer algebra.

We use just the elements of the theory of associative algebras. The reader can consult Herstein [16] and Pierce [26] (mainly Chapters 1–3) on this subject. Basic facts about Lie algebras (solvability, nilpotence, radicals) can be found in Humphreys [18] and Jacobson [20].

Polynomial time algorithms play a key role in the chapter. We refer to Hopcroft–Ullman [17], Chapters 12–13 and Cormen–Leiserson–Rivest [8] for an introduction to the complexity of algorithms.

The first polynomial time algorithm for computing the Jacobson radical over finite fields was given in [13]. Here we followed a novel approach proposed recently by Cohen–Ivanyos–Wales [5], where the problem is treated for arbitrary fields of characteristic $p > 0$.

The applications discussed in Section 4 to Lie algebras are from Rónyai [27]. The more recent [15] by de Graaf–Ivanyos–Rónyai also contains polynomial time methods for Lie algebras. Description of algorithms in the ELIAS package can be found in the Ph. D. thesis [14] of Willem de Graaf.

The algorithm presented here for the Wedderburn decomposition is from Friedl–Rónyai [13], where number fields are also considered as ground fields. A more efficient algorithm has been given recently by Eberly and Giesbrecht [11]. A solution for function fields over finite fields can be found in Ivanyos–Szántó–Rónyai [19].

The material for Section 6 and for Exercises 5.7–5.10 and 6.9–6.17 is mostly from Rónyai [28]. The problem of finding zero divisors over the rationals is considered in Rónyai [27]; [29] is a survey on computations in associative algebras.

References

1. E. H. Bareiss (1968): *Sylvester's identity and multistep integer-preserving Gaussian elimination*, Mathematics of Computation **103**, 565–578.
2. R. E. Beck, B. Kolman, and I. N. Stewart (1977): *Computing the structure of a Lie algebra*, Computers in nonassociative rings and algebras, Academic Press, New York, 167–188.
3. E. R. Berlekamp (1968): *Algebraic Coding Theory*, McGraw-Hill.
4. E. R. Berlekamp (1970): *Factoring polynomials over large finite fields*, Math. of Computation **24**, 713–715.
5. A. M. Cohen, G. Ivanyos, and D. B. Wales (1997): *Finding the radical of an algebra of linear transformations*, J. of Pure and Applied Algebra **117** & **118**, 177–193.
6. A. M. Cohen and L. Meertens (1995): *The ACELA project: Aims and Plans*, to appear in: Human Interaction for Symbolic Computation, ed. N. Kajler, Texts and Monographs in Symbolic Computation, Springer-Verlag, Berlin Heidelberg New York.
7. G. E. Collins, M. Mignotte, and F. Winkler (1983): *Arithmetic in basic algebraic domains*, in: Computer Algebra. Symbolic and Algebraic Computation, 2nd edn., Springer-Verlag, Berlin Heidelberg New York, 189–220.
8. T. H. Cormen, C. E. Leiserson, and R. L. Rivest (1990): *Introduction to Algorithms*, The MIT Press.
9. L. E. Dickson (1923): *Algebras and Their Arithmetics*, University of Chicago.
10. W. M. Eberly (1989): *Computations for Algebras and Group Representations*, Ph.D. thesis, Dept. of Computer Science, University of Toronto.
11. W. M. Eberly and M. Giesbrecht (1996): *Efficient decomposition of associative algebras*, Proc. of ISSAC'96, ACM Press, 170–178.
12. J. Edmonds (1967): *System of distinct representatives and linear algebra*, Journal of Research of the National Bureau of Standards **718**, 241–245.
13. K. Friedl and L. Rónyai (1985): *Polynomial time solution of some problems in computational algebra*, Proc. 17th ACM STOC, 153–162.
14. W. A. de Graaf (1997): *Algorithms for Finite-Dimensional Lie Algebras*, Ph.D. Thesis, Technische Universiteit Eindhoven.

15. W. A. de Graaf, G. Ivanyos, and L. Rónyai (1996): *Computing Cartan subalgebras of Lie algebras*, Applicable Algebra in Engineering, Communication and Computing **7**, 339–349.
16. I. N. Herstein (1968): *Noncommutative Rings*, Math. Association of America.
17. J. E. Hopcroft and J. D. Ullman (1979): *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley.
18. J. E. Humphreys (1980): *Introduction to Lie Algebra and Representation Theory*, Graduate Texts in Mathematics **9**, Springer-Verlag, Berlin Heidelberg New York.
19. G. Ivanyos, L. Rónyai, and Á. Szántó (1994): *Decomposition of algebras over* $\mathbb{F}_q(X_1, \ldots, X_m)$, Applicable Algebra in Engineering, Communication and Computing **5**, 71–90.
20. N. Jacobson (1962): *Lie Algebras*, John Wiley.
21. R. Kannan and A. Bachem (1979): *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, SIAM J. on Computing **4**, 499–507.
22. D. E. Knuth (1981): *The art of computer programming, Vol. 2, Seminumerical algorithms*, Addison-Wesley.
23. S. Lang (1965): *Algebra*, Addison-Wesley.
24. R. Lidl and H. Niederreiter (1983): *Finite Fields*, Addison-Wesley.
25. M. Mignotte (1992): *Mathematics for Computer Algebra*, Springer-Verlag, Berlin Heidelberg New York.
26. R. S. Pierce (1982): *Associative Algebras*, Springer-Verlag, Berlin Heidelberg New York.
27. L. Rónyai (1988): *Zero divisors in quaternion algebras*, Journal of Algorithms **9**, 494–506.
28. L. Rónyai (1990): *Computing the structure of finite algebras*, J. of Symbolic Computation **9**, 355–373.
29. L. Rónyai (1993): *Computations in associative algebras*, Groups and Computation, DIMACS Series **11**, American Mathematical Society, 221–243.

Chapter 6. Symbolic Recipes for Real Solutions

Laureano Gonzalez-Vega, Fabrice Rouillier, Marie-Françoise Roy, and Guadalupe Trujillo

1. Introduction

The main purpose of this chapter is to show how to use algorithms and methodology provided by computer algebra to manipulate in a symbolic way the real solutions of an algebraic system of equations.

For example, the solution of a concrete problem in neural networks (see [35]) is reduced to determine the real common solutions of the algebraic system of equations

$$\begin{aligned}1 - cx_1 - x_1x_2^2 - x_1x_3^2 &= 0\\ 1 - cx_2 - x_2x_1^2 - x_2x_3^2 &= 0\\ 1 - cx_3 - x_3x_1^2 - x_3x_2^2 &= 0,\end{aligned}$$

where c is a parameter taking only integer values. Two different, and very interesting, problems arise naturally from this example:

- first, for every instance of the parameter c to solve the corresponding polynomial system by determining, for example, the exact number of different real solutions and one approximation of the coordinates of every real solution, and
- second, to determine under which conditions on the parameter c the polynomial system has at least one real solution.

Describing some of the possibilities that computer algebra offers to answer such questions is the main goal of this chapter.

To answer the first question we are going to use some of the methods introduced in Chapter 2 since, from the information obtained about the complex solutions, we will recover information about the real solutions. This will be accomplished in two different ways: either by reducing the problem to a question in linear algebra (for example, the number of different real solutions will be equal to the signature of the trace matrix introduced in Chapter 2) or by reducing it to a univariate problem (for example, by using the Rational Univariate Representation introduced in Chapter 2, too). This last remark motivates the study in this chapter of the Real Root Counting Problem in the univariate case.

As for the second problem, by using techniques of Chapter 2, it will be reduced to determining the conditions the parameter c must verify in order that the polynomial

$$\begin{gathered}8cx_3^{14} - 8x_3^{13} + 28c^2x_3^{12} - 16cx_3^{11} + (38c^3+4)x_3^{10} - 2c^2x_3^9 + (25c^4+6c)x_3^8 \\ +(14c^3+6)x_3^7 + (8c^5+4c^2)x_3^6 + (10c^4+2c)x_3^5 + (c^6+c^3-5)x_3^4 \\ +(2c^5-3c^2)x_3^3 - cx_3^2 + (-2c^3+1)x_3 - c^2\end{gathered}$$

have a real solution. This is a typical Quantifier Elimination Problem and the last part of this chapter is devoted to showing how to solve it.

The different techniques of this chapter are illustrated in some concrete examples. We hope that we are presenting these techniques in such a way that they can be understood (and used) by a non-specialist in the subject; no more than a basic knowledge of elementary algebra (mainly linear algebra) is assumed. A prerequisite for the section devoted to the Real Root Counting Problem in the multivariate case, is Chapter 2.

The chapter is divided into five sections (disregarding this introduction and an appendix). The first two sections are devoted to the problem of counting real solutions for polynomial systems of equations. The first one, §2, gives a recent efficient method to determine the number of real roots of a univariate polynomial, based on Sylvester-Habicht sequences. The second section, §3, examines the same problem for polynomial systems of equations with a finite number of complex solutions, and presents Hermite's method and its variants.

The Sign Determination Scheme, to determine sign conditions realized by a finite set of polynomials on the real roots of a polynomial or a polynomial system, is studied in the third section. The fourth section contains a brief introduction to the characterization of real algebraic numbers via Thom's codes. In the final section, §6, it is shown how to perform Quantifier Elimination in several cases where the structure is simple enough to allow for direct application of the methods presented in the first two sections in a parameterized way.

An appendix, §7, proves the properties of the Sylvester-Habicht sequences in full detail.

2. Real Root Counting: The Univariate Case

This section presents some methods to count the real solutions of a univariate polynomial. We define the Cauchy index and relate it to real root counting. Then we describe an extension of the classical Sturm sequence, indicate the drawbacks of this method, and introduce a slight modification of the theory of subresultants, which exhibits a much better behaviour. This will be used in §6.

2.1 Computing the Number of Real Roots

First we define the Cauchy index of a rational function. If A is a univariate polynomial then, for the rational function A'/A, we shall see that its Cauchy index is equal to the number of real roots of A.

2.1.1 Cauchy Index and Real Root Counting. Let D be an integral domain contained in a real closed field R and let K be its field of fractions. Let A and B be univariate polynomials with coefficients in D.

The *multiplicity* of a root a of A in R is, as usual, the exponent k such that $A = (X-a)^k A_1$ with A_1 a polynomial such that $A_1(a) \neq 0$. The *polar multiplicity* of a in the rational function B/A is the integer k such that

$$\frac{B}{A} = \frac{B_1}{(X-a)^k A_1}$$

with $X-a$ dividing neither B_1 nor A_1. The rational function B/A has a finite limit at a when the polar multiplicity of a in B/A is nonpositive. The rational function B/A has an infinite limit at a_- (respectively, a_+) if its polar multiplicity at a is strictly positive. The limit at a_+ is $\sigma\infty$ where $\sigma = \operatorname{sign}(B_1(a)/A_1(a))$. The limit at a_- is $\sigma\infty$ where $\sigma = (-1)^k \operatorname{sign}(B_1(a)/A_1(a))$.

Definition 2.1. The *Cauchy index* of B/A, denoted by $I(B/A)$, is the number of jumps of the function B/A from $-\infty$ to $+\infty$ minus the number of jumps of the function B/A from $+\infty$ to $-\infty$. In other words,

$$I(B/A) = \sum_{\{a \in \mathrm{R} \mid A(a)=0\}} \varepsilon_a,$$

where ε_a is defined as follows.

$$\varepsilon_a = \begin{cases} +1 & \text{if } \lim\limits_{x\to a_-} \dfrac{B}{A} = -\infty \text{ and } \lim\limits_{x\to a_+} \dfrac{B}{A} = +\infty \\ -1 & \text{if } \lim\limits_{x\to a_-} \dfrac{B}{A} = +\infty \text{ and } \lim\limits_{x\to a_+} \dfrac{B}{A} = -\infty \\ 0 & \text{otherwise.} \end{cases}$$

The connection between the Cauchy index and the number of roots in a real closed field is given by the corollary to the next proposition. Let

$$\begin{aligned} \mathrm{c}(A) &= \operatorname{card}\{x \in R \mid A(x) = 0\}, \\ \mathrm{c}_{[+]}(A;B) &= \operatorname{card}\{x \in R \mid A(x) = 0\,,\ B(x) > 0\}, \\ \mathrm{c}_{[0]}(A;B) &= \operatorname{card}\{x \in R \mid A(x) = 0\,,\ B(x) = 0\}, \\ \mathrm{c}_{[-]}(A;B) &= \operatorname{card}\{x \in R \mid A(x) = 0\,,\ B(x) < 0\}. \end{aligned}$$

Proposition 2.2. $I(A'B/A) = \mathrm{c}_{[+]}(A;B) - \mathrm{c}_{[-]}(A;B)$.

Proof. We restrict our attention to the roots c of A which are not roots of B since $A'B/A$, at a common root of A and B, has a finite limit. Defining k as the multiplicity of the root c in A, we find

$$\frac{A'B}{A} = \frac{kB(c)}{X-c} + R_c$$

with R_c having a finite value at c; so it is easy to see that

- o there is a jump from $-\infty$ to $+\infty$ at c in $A'B/A$ ($\varepsilon_c = 1$) if $B(c) > 0$,
- o there is a jump from $+\infty$ to $-\infty$ at c in $A'B/A$ ($\varepsilon_c = -1$) if $B(c) < 0$.

Corollary 2.3. $I(A'/A) = c(A)$.

The Cauchy index is also used in the solution of the Routh-Hurwitz problem, i.e., to determine the number of different complex roots with real negative part of a polynomial with real coefficients (see [16]).

2.2 Sylvester Sequence

2.2.1 Computing the Sylvester Sequence.

The *signed remainder sequence*, or *Sylvester sequence*, of A and B is defined as follows.

Algorithm 2.4 (Sy).
Input: The polynomials A and B.
Output: The Sylvester sequence

$$Sy^j = \mathrm{Sy}^j(A,B).$$

Initialization: $Sy^0 := A$, $Sy^1 := B$.
Loop: ($i = 2, \ldots$) The polynomials Sy^{i-2} and Sy^{i-1} are already known. The polynomial Sy^i will be computed.

$$Sy^i = -\mathrm{Rem}(Sy^{i-2}, Sy^{i-1})$$

End: The algorithm ends with Sy^w, the (signed) gcd of A and B when $Sy^{w+1} = 0$.

When $B = A'$, we recover the usual *Sturm sequence* of A: $\mathrm{Stu}^j(A) = \mathrm{Sy}^j(A, A')$.

Definition 2.5. The *usual case* is the case where the degree of A is d, the degree of B is $d-1$, and the degrees of the polynomials in the remainder sequence decrease exactly by one. Note that, in the usual case, the degree of Sy^{d-j} is exactly j.

Definition 2.6. The *i-th quotient Q_i in the signed Euclidean division* of A by B, is defined as

$$Sy^{i-2} = Q_i \cdot Sy^{i-1} - Sy^i.$$

The *signed Euclidean transition matrix* is

$$\begin{pmatrix} 0 & 1 \\ -1 & Q_{i-1} \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ -1 & Q_0 \end{pmatrix}.$$

It satisfies

$$\begin{pmatrix} 0 & 1 \\ -1 & Q_{i-1} \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ -1 & Q_0 \end{pmatrix} \cdot \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} Sy^{i-1} \\ Sy^i \end{pmatrix}.$$

Note that this matrix is *unimodular* (that is, it has determinant equal to 1).

Below we give the Sturm sequence of the polynomial

$$\begin{aligned} A := 9X^{13} - 18X^{11} - 33X^{10} + 102X^8 + 7X^7 - 36X^6 \\ -122X^5 + 49X^4 + 93X^3 - 42X^2 - 18X + 9; \end{aligned}$$

It also is the Sylvester sequence of A and A'.

$$\begin{aligned} \mathrm{Stu}^2(A) = {} & \tfrac{1}{13}(36X^{11} + 99X^{10} - 510X^8 - 42X^7 + 252X^6 + 976X^5 \\ & -441X^4 - 930X^3 + 462X^2 + 216X - 117) \end{aligned}$$

$$\begin{aligned} \mathrm{Stu}^3(A) = {} & \tfrac{1}{16}(10989X^{10} + 21240X^9 - 70746X^8 - 6054X^7 - 13932X^6 \\ & +159044X^5 - 24463X^4 - 153878X^3 + 59298X^2 + 35628X \\ & -17019) \end{aligned}$$

$$\begin{aligned} \mathrm{Stu}^4(A) = {} & \tfrac{32}{1490841}(626814X^9 - 1077918X^8 + 71130X^7 - 830472X^6 \\ & +2259119X^5 + 460844X^4 - 2552804X^3 + 668517X^2 \\ & +632094X - 256023) \end{aligned}$$

$$\begin{aligned} \mathrm{Stu}^5(A) = {} & \tfrac{165649}{38804522528}(43475160X^8 - 57842286X^7 + 5258589X^6 \\ & -92294719X^5 + 134965334X^4 + 31205119X^3 \\ & -79186035X^2 + 5258589X + 9147321) \end{aligned}$$

$$\begin{aligned}\mathrm{Stu}^6(A) = {} & \tfrac{2425282658}{543561530761725025}(1584012126X^7 - 2548299819X^6 \\ & +984706749X^5 - 3696028294X^4 + 5946032911X^3 \\ & -713636955X^2 - 2548299819X + 984706749)\end{aligned}$$

$$\begin{aligned}\mathrm{Stu}^7(A) = {} & \tfrac{543561530761725025}{6761403525275795353156 96712}(12232018869X^6 - 8633929833X^5 \\ & -28541377361X^3 + 20145836277X^2 + 12232018869X \\ & -8633929833)\end{aligned}$$

$$\mathrm{Stu}^8(A) = -\tfrac{6670589018492723310272156415951 4728}{180730930229098050132455395887141 5645}(3X^5 - 7X^2 + 3) .$$

Note that the size of the coefficients appearing in the Sturm sequence is big. It has been shown recently (see [30] or [31]) that it is quadratic in the degree of the input polynomial.

When dealing with Sturm (Sylvester) sequences there are also *specialization problems*. Let D be a domain, K its field of fractions, A and B polynomials in $D[X]$. Suppose that the computation of the Sylvester sequence has been done in the field K, and that the coefficients of A and B are *specialized* (that is, we consider a ring morphism f from D to a domain D' and its natural images $f(A)$ and $f(B)$ under the extension $f\colon D[X] \to D'[X]$). The Sylvester sequence associated to $f(A)$ and $f(B)$ is not easy to compute from the Sylvester sequence of A and B because, in the Euclidean division process of A by B, elements of D appear in the denominator, and may well specialize to 0. In this case the Sylvester sequence of $f(A)$ and $f(B)$ is not obtained by specializing the Sylvester sequence of A and B, and the degree of the polynomials in the Sylvester sequence of $f(A)$ and $f(B)$ does not agree with the degree of the polynomials in the Sylvester sequence of A and B.

Example 2.7. Consider the general polynomial of degree 4,

$$A = X^4 + pX^2 + qX + r.$$

The Sturm sequence of A computed in $\mathbb{Q}(p,q,r)[X]$ is

$$\mathrm{Stu}^0(A) = X^4 + pX^2 + qX + r$$

$$\mathrm{Stu}^1(A) = 4X^3 + 2pX + q$$

$$\mathrm{Stu}^2(A) = \frac{-(2pX^2 + 3qX + 4r)}{4}$$

$$\mathrm{Stu}^3(A) = \frac{-((2p^3 - 8pr + 9q^2)X + p^2q + 12qr)}{p^2}$$

$$\mathrm{Stu}^4(A) = \frac{p^2(16p^4r - 4p^3q^2 - 128p^2r^2 + 144pq^2r - 27q^4 + 256r^3)}{4(2p^3 - 8pr + 9q^2)^2}$$

When we choose particular values $\tilde{p}$, $\tilde{q}$, $\tilde{r}$ for p, q, r, then the Sturm sequence of $\tilde{A} = X^4 + \tilde{p}X^2 + \tilde{q}X + \tilde{r}$ is generally obtained by replacing p, q, r by $\tilde{p}$, $\tilde{q}$, $\tilde{r}$, respectively, in the Sturm sequence of $X^4 + pX^2 + qX + r$. But in some cases (when denominators vanish) this substitution is impossible and the computation has to be redone. For $\tilde{p} = 0$, the Sturm sequence of $\tilde{A} = X^4 + \tilde{q}X + \tilde{r}$ is:

$$\mathrm{Stu}^0(\tilde{A}) = X^4 + \tilde{q}X + \tilde{r}$$

$$\mathrm{Stu}^1(\tilde{A}) = 4X^3 + \tilde{q}$$

$$\mathrm{Stu}^2(\tilde{A}) = -(3\tilde{q}X + 4\tilde{r})/4$$

$$\mathrm{Stu}^3(\tilde{A}) = -(27\tilde{q}^4 - 256\tilde{r}^3)/(27\tilde{q}^3).$$

2.2.2 Sylvester Sequence and Cauchy Index. The Cauchy index can be computed from the Sylvester sequence as follows. Let A and B be polynomials in $D[X]$. We denote by

$$V_{\mathrm{Sy}}(A, B; a) = V(\{\mathrm{Sy}^j(A, B)\}_{j=0,\ldots,d}; a)$$

the number of sign changes of the Sylvester sequence of A and B at $a \in R \cup \{-\infty, +\infty\}$, and put

$$V_{\mathrm{Sy}}(A, B) = V_{\mathrm{Sy}}(A, B; -\infty) - V_{\mathrm{Sy}}(A, B; +\infty).$$

Theorem 2.8. $V_{\mathrm{Sy}}(A, B) = I(B/A)$.

A proof of this result can be found for example in [42].

Corollary 2.9. $V_{\mathrm{Sy}}(A, A'B) = \mathrm{c}_{[+]}(A; B) - \mathrm{c}_{[-]}(A; B)$.

Corollary 2.10. *Let A be a polynomial in $D[X]$. Then $V_{\mathrm{Sy}}(A, A')$ is the number of real roots of A.*

2.3 Sylvester-Habicht Sequence

The Sylvester-Habicht sequence gives, just like the Sylvester sequence, the Cauchy index of a rational function, but with a better control of the size of coefficients and without specialization problems.

In the following, we are going to define the Sylvester-Habicht sequence by means of an algorithm. We first indicate the algorithm in the usual case, so that the analogy with the signed remainder sequence becomes apparent; then we give the algorithm for the general case.

An alternative definition of the polynomials in the Sylvester-Habicht sequence, together with the correctness proof of the algorithm, appears in the appendix to this chapter, §7.

2.3.1 Computing the Sylvester-Habicht Sequence. For A of degree d, we take

$$\begin{array}{rcl} A & = & a_dX^d + a_{d-1}X^{d-1} + a_{d-2}X^{d-2} + \cdots + a_0, \\ B & = & b_qX^q + \cdots + b_0. \end{array}$$

The Sylvester-Habicht sequence of A and B consists, for $0 \leq j \leq d$, of polynomials $H_j(A,B)$ of respective degrees $\leq j$. In the usual case, $H_j(A,B)$ is of degree j. The j-th principal Sylvester-Habicht coefficient, which is the coefficient of degree j of H_j, will be denoted by

$$h_j = \mathrm{syha}_j(A,B).$$

By convention, $H_d(A,B) = A$.

The following algorithm computes the Sylvester-Habicht sequence in the usual case.

Algorithm 2.11 (SyHaU).
Input: Polynomials A and B of respective degrees d and $d-1$.
Output: The Sylvester-Habicht sequence

$$H_j = \mathrm{SyHa}_j(A,B)$$

with $0 \leq j \leq d$.
Initialization: $d := \deg(A)$; $H_d := A$; $H_{d-1} := B$;

$$H_{d-2} := -\mathrm{Rem}(h_{d-1}^2 H_d, H_{d-1});$$

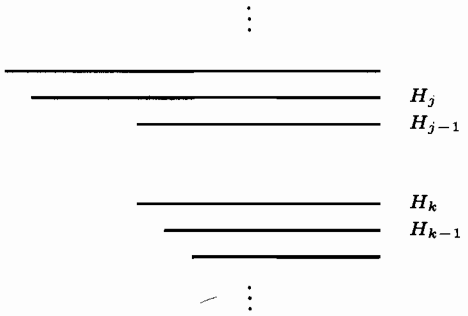
Loop: $(j < d)$ The polynomials H_j, H_{j-1}, and h_j are already known with h_j nonzero and $j-1 = \deg(H_{j-1})$. The polynomial H_{j-2} will be computed:

$$H_{j-2} := -\frac{1}{h_j^2}\mathrm{Rem}(h_{j-1}^2 H_j, H_{j-1}).$$

End: The algorithm ends when H_0 has been computed, i.e., when $j \leq 1$.

***Remark 2.12.* In the usual case above, it is clear that H_j is proportional to Sy^{d-j}, and that the ratio is a square.**

In the general case, the Sylvester-Habicht polynomials will present the famous gap structure, graphically displayed by the following diagram of Habicht lines.



More precisely, the following algorithm computes them.

Algorithm 2.13 (SyHa).

Input: The polynomials A and B with $d = \deg(A)$.
Output: The Sylvester-Habicht sequence

$$H_j = \mathrm{SyHa}_j(A, B)$$

with $0 \leq j \leq d$. The principal Sylvester-Habicht coefficients will be denoted by

$$h_j = \mathrm{syha}_j(A, B).$$

Initialization: $H_d := A$, $\overline{h}_d := {a_d}^{-1}$, and

$$H_{d-1} := \begin{cases} B & \text{if } q = \deg(B) < d, \\ \mathrm{Rem}(a_d^{2e}B, A) & \text{otherwise, where } e = \lceil \frac{q-d+1}{2} \rceil \end{cases}$$

Loop: H_j, $\overline{h}_j$, and H_{j-1} are already known with $h_j = \overline{h}_j$ nonzero and $k = \deg(H_{j-1})$. The lacking H_ℓ and $\overline{h}_\ell$ are going to be computed up to H_{k-1} and $\overline{h}_k$.

1. Computation of H_ℓ for $k < \ell < j - 1$.
   If $k < j - 2$, then
   $$H_\ell = 0.$$
2. Computation of $\overline{h}_\ell$ for $k < \ell < j - 1$.
   Define c_{j-1} as the leading coefficient of H_{j-1}. If $k = j - 1$, then there is nothing to compute; else ($k < j - 1$) compute, for ℓ decreasing from $j - 1$ to k, $\overline{h}_\ell$ by $\overline{h}_{j-1} := c_{j-1}$, and
   $$\overline{h}_\ell := (-1)^{j-\ell-1} \frac{\overline{h}_{\ell+1} c_{j-1}}{\overline{h}_j}.$$

3. Computation of H_k.
   Take
$$H_k := \frac{\overline{h}_k H_{j-1}}{\mathrm{c}_{j-1}}, \quad h_k := \overline{h}_k.$$
4. Computation of H_{k-1}.
$$H_{k-1} := -\frac{1}{h_j \overline{h}_j} \mathrm{Rem}(\mathrm{c}_{j-1} h_k H_j, H_{j-1}).$$

End: The algorithm ends when H_0 has been computed, i.e., when $j \leq 1$.

Remark 2.14. Note that $h_d \overline{h}_d = 1$ and $h_j \overline{h}_j = h_j^2$ when $j < d$ and $\deg(H_j) = j$.

The values of j with $h_j \neq 0$ are precisely the degrees of the polynomials in the signed remainder sequence. Note that the algorithm for the general case contains, as a particular case, the algorithm for the usual case since in that situation $h_d \overline{h}_d = 1$, $H_{j-1} = H_k, \mathrm{c}_{j-1} = h_k = h_{j-1}$. We gave the algorithm for the usual case only to stress the analogy between the Sylvester-Habicht sequence and the signed remainder sequence.

Remark 2.15. The algorithm should be executed as follows (for simplicity, we do not take signs into account in what follows).

- For the computation of $\overline{h}_k$, take c_{j-1} and make $j-k-1$ times the following computation: multiply by c_{j-1} and divide by $\overline{h}_j$. All the intermediate divisions are exact, i.e., with a result in $D[X]$. In particular, for $j = d$, dividing by $\overline{h}_d$ is multiplying by a_d.
- To compute H_k it suffices to multiply H_{j-1} by $\overline{h}_{k+1}$ and divide the result by h_j (up to sign).
- For the computation of H_{k-1}, take H_j, multiply it by c_{j-1}, then by h_k, perform the Euclidean division of the polynomial thus obtained by H_{j-1} (the quotient and remainder are in $D[X]$), and divide the result by $h_j \overline{h}_j$.

In the appendix to this chapter (§7), we prove that all the divisions needed in the algorithm are exact, and that all the intermediate steps of the computation above take place in $D[X]$. This is proved for the computation of $\overline{h}_k$ in Lemma 7.10. The fact that the Euclidean remainder of the division of $\mathrm{c}_{j-1} h_k H_j$ by H_{j-1} can be done entirely in D follows from Corollary 7.18.

When $B = A'$, we recover the so-called *Sturm-Habicht sequence* of A: $\mathrm{StHa}^j(A) = \mathrm{SyHa}_j(A, A')$ (see [18] or [19]). The next example shows the Sturm-Habicht sequence of the polynomial

$$A := 9X^{13} - 18X^{11} - 33X^{10} + 102X^8 + 7X^7 - 36X^6$$
$$-122X^5 + 49X^4 + 93X^3 - 42X^2 - 18X + 9,$$

i.e., the Sylvester-Habicht sequence of A and A'. The sequence is:

$$\mathrm{StHa}_0(A) = \mathrm{StHa}_1(A) = \mathrm{StHa}_2(A) = \mathrm{StHa}_3(A) = \mathrm{StHa}_4(A) = 0,$$

$$\mathrm{StHa}_5(A) = -5503923710091 2075040(3X^5 - 7X^2 + 3)$$

$$\begin{aligned}\mathrm{StHa}_6(A) = {} & -12397455648(12232018869X^6 - 8633929833X^5 \\ & -28541377361X^3 + 20145836277X^2 + 12232018869X \\ & -8633929833)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_7(A) = {} & -1377495072(1584012126X^7 - 2548299819X^6 \\ & +984706749X^5 - 3696028294X^4 + 5946032911X^3 \\ & -713636955X^2 - 2548299819X - 2548299819X \\ & +984706749)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_8(A) = {} & -38263752(43475160X^8 - 57842286X^7 + 5258589X^6 \\ & -92294719X^5 + 134965334X^4 + 31205119X^3 \\ & -79186035X^2 + 5258589X + 9147321)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_9(A) = {} & -1062882(626814X^9 - 1077918X^8 + 71130X^7 - 830472X^6 \\ & +2259119X^5 + 460844X^4 - 2552804X^3 + 668517X^2 \\ & +632094X - 256023)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_{10}(A) = {} & -6561(10989X^{10} + 21240X^9 - 70746X^8 - 6054X^7 \\ & -13932X^6 + 159044X^5 - 24463X^4 - 153878X^3 \\ & +59298X^2 + 35628X - 17019)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_{11}(A) = {} & 1053(36X^{11} + 99X^{10} - 510X^8 - 42X^7 + 252X^6 + 976X^5 \\ & -441X^4 - 930X^3 + 462X^2 + 216X - 117)\end{aligned}$$

$$\begin{aligned}\mathrm{StHa}_{12}(A) &= A' \\ \mathrm{StHa}_{13}(A) &= A.\end{aligned}$$

Note that the size of the coefficients is moderate, compared with the Sturm sequence. It will be a consequence of the definition of the Sylvester-Habicht sequence to be given in the appendix to this chapter that their bit size is linear in the degree d of the considered polynomials (cf. Corollary 7.8).

Given a ***specialization***, i.e., a ring morphism $f\colon D \longrightarrow D'$, we are interested in an easy way to compute the Sylvester-Habicht sequence of $f(A)$ and $f(B)$ when the Sylvester-Habicht sequence of A and B is known.

Proposition 2.16. *Let $f\colon D \to D'$ be a ring homomorphism such that $d = \deg(A) = \deg(f(A))$, $d > q = \deg(B)$ (so that $q \geq \deg(f(B))$), then, for all $j \leq d$,*

$$\mathrm{SyHa}_j(f(A), f(B)) = f(\mathrm{SyHa}_j(A, B)).$$

The specialization properties of the Sylvester-Habicht sequence are shown on the general polynomial of degree 4:

$$A = X^4 + pX^2 + qX + r.$$

The Sturm-Habicht sequence of A is formed by the polynomials (belonging to $\mathbb{Z}[p, q, r][X]$):

$$\begin{aligned}
\mathrm{StHa}_4(A) &= X^4 + pX^2 + qX + r\\
\mathrm{StHa}_3(A) &= 4X^3 + 2pX + q\\
\mathrm{StHa}_2(A) &= -4(2pX^2 + 3qX + 4r)\\
\mathrm{StHa}_1(A) &= -4((2p^3 - 8pr + 9q^2)X + p^2q + 12qr)\\
\mathrm{StHa}_0(A) &= 16p^4r - 4p^3q^2 - 128p^2r^2 + 144pq^2r - 27q^4 + 256r^3.
\end{aligned}$$

It agrees, up to squares in $\mathbb{Q}(p, q, r)$, with the generic Sturm sequence for A. If $p = 0$, the Sturm-Habicht sequence of the polynomial $A = X^4 + qX + r$ is

$$\begin{aligned}
\mathrm{StHa}_4(A) &= X^4 + qX + r\\
\mathrm{StHa}_3(A) &= 4X^3 + q\\
\mathrm{StHa}_2(A) &= -4(3qX + 4r)\\
\mathrm{StHa}_1(A) &= -12q(3qX + 4r)\\
\mathrm{StHa}_0(A) &= -27q^4 + 256r^3.
\end{aligned}$$

which is the specialization of the Sturm-Habicht sequence of A with $p = 0$. As we have seen above, this was not the case with Sturm sequences.

Finally, remark that, although it may not be clear from the presentation of the Sylvester-Habicht sequence given here, the polynomials in this sequence always belong to $D[X]$ and denominators never appear. This is explained in the appendix (see §7.1), where the coefficients of the polynomials in the Sylvester-Habicht sequence are shown to be determinants of matrices whose entries are coefficients of the initial polynomials. So much for the reason why the use of Sturm-Habicht sequences avoids the specialization problems attached to Sturm sequences.

2.3.2 Sylvester-Habicht Sequence and Cauchy Index. We are going to explain that the sign variations in the Sylvester-Habicht sequence give the Cauchy index.

Let A and B be polynomials in $D[X]$. We denote by $V_{\mathrm{SyHa}}(A, B; a) = V(\{\mathrm{SyHa}_j(A, B)\}_{j=0,\dots,d}; a)$ the number of sign changes of the Sylvester-Habicht sequence at a and define

$$V_{\mathrm{SyHa}}(A, B) = V_{\mathrm{SyHa}}(A, B; -\infty) - V_{\mathrm{SyHa}}(A, B; +\infty).$$

Theorem 2.17. $V_{\text{SyHa}}(A, B) = I(B/A)$.

The proof of this result appears in §7, the appendix to this chapter. Note that in the usual case, Theorem 2.17 is an immediate consequence of Theorem 2.8, because of Remark 2.12.

Corollary 2.18. $V_{\text{SyHa}}(A, A'B) = c_{[+]}(A; B) - c_{[-]}(A; B)$.

Corollary 2.19. *Let A be a polynomial in $D[X]$. Then $V_{\text{SyHa}}(A, A')$ is the number of real roots of A.*

Next we indicate how to compute the Cauchy index by using only the principal Sylvester-Habicht coefficients, rather than the whole Sylvester-Habicht sequence. First we need a definition. Given an injective mapping $\ell : [0, \ldots, d] \rightarrow \mathbb{N}$, we consider an ordered list $[a]$ of nonzero elements of an ordered field K indexed by ℓ. We define $D([a])$ as

$$\sum_{i=0}^{d-1} \sigma(i)$$

where

$$\sigma(i) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } \ell(i+1) - \ell(i) \text{ is even} \\ (-1)^{(\ell(i+1)-\ell(i)-1)/2} \cdot \operatorname{sign}(a_{\ell(i)} a_{\ell(i+1)}) & \text{if } \ell(i+1) - \ell(i) \text{ is odd} \end{cases}$$

Note that when $\ell(i) = i$ for every i, the integer $D([a])$ is nothing but the difference between the number of sign changes in $[a_0, \ldots, (-1)^i a_i, \ldots, (-1)^d a_d]$ and the number of sign changes in $[a_0, \ldots, a_i, \ldots, a_d]$. It can be described also as the difference between the number of sign permanences and the number of sign changes in $[a_0, \ldots, a_d]$.

Proposition 2.20. *Let A and B be polynomials in $D[X]$. We denote by ℓ the finite sequence of integers such that $\text{SyHa}_j(A, B)$ is not defective, i.e., $\text{syha}_j(A, B) \neq 0$, and $[h]$ the corresponding list of $h_j = \text{syha}_j(A, B)$. Then*

$$D([h]) = I(B/A).$$

The proof appears in §7, the appendix to this chapter.

2.4 Some Recipes for Counting Real Roots

The Cauchy index can be computed using Algorithm **SyHa** and Proposition 2.20. We give recipes to determine the number of real roots of a univariate polynomial or the Sturm Query of A and B, denoted by $SQ(A, B)$, which is the integer $c_{[+]}(A; B) - c_{[-]}(A; B)$.

Recipe $\mathtt{CRS}_1$: Counting real solutions for a polynomial A.

- Compute the principal coefficients of the Sturm-Habicht sequence of A using Algorithm **SyHa** applied to A and A': $[h] = [h_d, \dots, h_0]$.
- Compute $D([h])$, which is the number of real roots of A.

Recipe $\mathtt{SQ}_1$: Sturm Query of A and B: $\mathrm{c}_{[+]}(A;B) - \mathrm{c}_{[-]}(A;B)$.

- Compute the principal coefficients of the Sylvester-Habicht sequence of A and $A'B$ by use of Algorithm **SyHa**: $[h] = [h_d, \dots, h_0]$.
- Compute $D([h])$, which is the Sturm Query of A and B.

Another method for computing the number of real roots of a univariate polynomial is Uspensky's method, based on Descartes' rule (see [10]).

3. Real Root Counting: The Multivariate Case

In Chapter 2, several problems regarding the complex solutions of a polynomial system of equations were reduced to linear algebra problems with the help of Gröbner Bases computations. In this section we continue with the same philosophy but now we are interested in information on real solutions. Let R be a real closed field containing a field K, let $\overline{K}$ be an algebraically closed field with $R \subset \overline{K}$, and let $\mathcal{P} = \{P_1, \dots, P_m\}$ be a finite set of polynomials with coefficients in K. Suppose that $A = K[X_1, \dots, X_k]/\mathcal{I}(\mathcal{P})$ is a finite dimensional vector space over K. Let N be the dimension of A as a K-vector space and n be the number of distinct solutions of $\mathcal{P}$ in $\overline{K}^n$, so that $n \leq N$. Let $\mathcal{Z}_R(\mathcal{P})$ denote the set of real solutions of $\mathcal{P}$. Given $h \in A$, we define, the *Sturm Query* of h with respect to $\mathcal{P}$ by

$$SQ(\mathcal{P}, h) = \#\{x \in \mathcal{Z}_R(\mathcal{P}) \mid h(x) > 0\} - \#\{x \in \mathcal{Z}_R(\mathcal{P}) \mid h(x) < 0\}.$$

In particular, when $h = 1$, we find the number of real solutions of $\mathcal{P}$: the cardinality of $\mathcal{Z}_R(\mathcal{P})$.

We associate to h the quadratic form, called *trace form*, defined in §3 of Chapter 2,

$$\begin{array}{rccc} Q_h\colon & A & \longrightarrow & K \\ & f & \longmapsto & \mathrm{Tr}(L_{f^2h}) \end{array}$$

and the corresponding bilinear form TrB_h. The *signature* of a quadratic form is the difference of the numbers of positive and negative entries once the quadratic form is diagonalized.

Theorem 3.1 ([2, 3, 14, 36, 37]).

$$SQ(\mathcal{P}, h) = \mathrm{signature}(Q_h).$$

Proof. Choose a separating element $u \in A$. If

$$\sum_{i=0}^{n-1} \lambda_i u^i$$

is a linear combination of powers of u which is 0 in A, then

$$P(T) = \sum_{i=0}^{n-1} \lambda_i T^i$$

has n distinct roots. Thus, the elements $1, u, \ldots, u^{n-1}$ are linearly independent in A, as $u(\mathcal{Z}_R(\mathcal{P}))$ has n distinct elements. Consider a basis

$$\omega_1 = 1, \omega_2 = u, \ldots, \omega_n = u^{n-1}, \omega_{n+1}, \ldots, \omega_N$$

of A. Then, given $(f_1, \ldots, f_N)$, denote by

$$f = \sum_{i=1}^{N} f_i \varpi_i$$

the corresponding linear combination. According to Stickelberger Theorem (see Chapter 2), on this basis, $Q_h(f)$ is given by

$$\sum_{\alpha \in \mathcal{Z}_R(\mathcal{P})} \mu_\alpha h(\alpha) \Big(\sum_{i=1}^{N} f_i \omega_i(\alpha) \Big)^2 + \sum_{\alpha \in \mathcal{Z}_{\overline{K}}(\mathcal{P}) \setminus \mathcal{Z}_R(\mathcal{P})} \mu_\alpha h(\alpha) \Big(\sum_{i=1}^{N} f_i \omega_i(\alpha) \Big)^2$$

as a quadratic form of the variables f_i. The signature of

$$\sum_{\alpha \in \mathcal{Z}_{\overline{K}}(\mathcal{P}) \setminus \mathcal{Z}_R(\mathcal{P})} h(\alpha) \Big(\sum_{i=1}^{N} f_i \omega_i(\alpha) \Big)^2$$

is 0 since, for α and $\overline{\alpha}$ complex conjugate solutions of $\mathcal{P}$,

$$h(\alpha) \Big(\sum_{i=1}^{N} f_i \omega_i(\alpha) \Big)^2 + h(\overline{\alpha}) \Big(\sum_{i=1}^{N} f_i \omega_i(\overline{\alpha}) \Big)^2$$

is a difference of two real squares. So the signature of Q_h agrees with the signature of

$$\sum_{\alpha \in \mathcal{Z}_R(\mathcal{P})} \mu_\alpha h(\alpha) \Big(\sum_{i=1}^{N} f_i \omega_i(\alpha) \Big)^2,$$

which is obviously equal to $SQ(\mathcal{P}, h)$.

The signature can be computed, for example, as follows.

Proposition 3.2. *If S is a symmetric $N \times N$ matrix with entries in R and*

$$P_S(\lambda) = a_0 + a_1 \lambda + \cdots + (-1)^N \lambda^N$$

is the characteristic polynomial of S, then the signature of S is equal to the difference of the number of sign variations and the number of sign permanences in $\{a_0, a_1, \ldots, (-1)^N\}$.

This method was used in [48] and [37]. More efficient methods appear in [41].

The preceding results produce the first recipe for determining directly the number of real solutions of the considered system (the Sturm Query for $\mathcal{P}$ and 1).

Recipe CRS$_2$: Counting real solutions for a zero-dimensional polynomial system $\mathcal{P}$.

- Compute the matrix of Q_1 (or TrB$_1$) as in §3 of Chapter 2, through a Gröbner basis computation and several normal form determinations.
- Compute the signature of Q_1, which is the number of different real solutions of $\mathcal{P}$.

Recipe SQ$_2$: Counting the Sturm Query of h for a zero-dimensional polynomial system $\mathcal{P}$.

- Compute the matrix of Q_h (or TrB$_h$) as in §3 of Chapter 2, through a Gröbner basis computation and several normal form determinations.
- Compute the signature of Q_h, which is $SQ(\mathcal{P}, h)$.

When the number of equations is equal to the number of unknowns, there is another recipe determining the number of real solutions or the Sturm Query.

Recipe CRS$_3$ (respectively, Recipe SQ$_3$).
Determine the Bezoutian of $P_1, \ldots, P_k$, Bez by computing the determinant $B(\underline{X}, \underline{Y})$ introduced in §4 of Chapter 2 and its Normal Form in $A \otimes A$:

$$\mathrm{Bez} = \sum_{\omega,\omega' \in \mathcal{A}} a_{\omega,\omega'} \omega(\underline{X}) \omega'(\underline{Y})$$

where $\mathcal{A}$ is the basis of A as K-vector space. If $\mathcal{B}$ is the matrix of the $a_{\omega,\omega'}$'s, then signature$(\mathcal{L}_{\mathrm{Jac}} \cdot \mathcal{B}) = \#(\mathcal{Z}_R(\mathcal{P}))$ (respectively, signature$(\mathcal{L}_{\mathrm{Jac}\cdot h} \cdot \mathcal{B}) = SQ(\mathcal{P}, h)$) where $\mathcal{L}_{\mathrm{Jac}}$ (respectively, $\mathcal{L}_{\mathrm{Jac}\cdot h}$) is the matrix of L_{Jac} (respectively, $L_{\mathrm{Jac}\cdot h}$) with respect to the basis $\mathcal{A}$.

Example 3.3. Let us consider the polynomial system of equations:

$$P_1 := X_1^2 X_2 - 2X_1^2 + X_2^2 + X_1 X_2 = 0, \qquad P_2 := 2X_1^2 - X_2^2 + X_1 X_2 = 0$$

already considered in Chapter 2. In this example, the R-basis of A is:

$$\mathcal{A} = \{1, X_2, X_2^2, X_2^3, X_1, X_1 X_2\} = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}.$$

The computations performed in Chapter 2 produce the matrix of Q_1 with respect to $\mathcal{A}$:

$$\mathrm{Tr} = \begin{pmatrix} 6 & -2 & 20 & -56 & -4 & 4 \\ -2 & 0 & -56 & 272 & 4 & -40 \\ 20 & -56 & 272 & -992 & -40 & 112 \\ -56 & 272 & -992 & 4160 & 112 & -544 \\ -4 & 4 & -40 & 112 & 8 & -8 \\ 4 & -40 & 112 & -544 & -8 & 80 \end{pmatrix},$$

which allows us to conclude that signature(Tr) $= 3$; thus the polynomial system of equations has exactly three real solutions.

In order to use **Recipe CRS**$_3$, first $B(\underline{X},\underline{Y})$ is computed:

$$\begin{aligned} B(\underline{X},\underline{Y}) = {} & -4X_1Y_1 - 2X_2Y_2 - Y_1X_2^2 - Y_1X_2Y_2 - X_1X_2Y_2 + X_1X_2Y_1 \\ & -2Y_1^2X_1 - 2X_2^2 - X_1X_2^2 - 2Y_1^3 - 4Y_1^2. \end{aligned}$$

The matrix $\mathcal{B}$ is obtained through the computation of the normal form of $B(\underline{X},\underline{Y})$:

$$\mathcal{B} = \begin{pmatrix} 0 & 0 & -2 & -1 & 0 & -4 \\ 0 & -2 & 0 & 0 & 0 & -1 \\ -2 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -4 & 1 \\ -4 & -1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Finally, $\mathcal{L}_{\mathrm{Jac}}$ is determined

$$\mathcal{L}_{\mathrm{Jac}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -8 & 0 & 0 & 0 & 0 & 0 \\ -6 & 2 & -20 & 56 & 4 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -26 & -8 & -64 & 64 & 20 & 16 \end{pmatrix},$$

and the use of the previous recipe allows us to conclude that

$$\mathrm{signature}(\mathcal{L}_{\mathrm{Jac}} \cdot \mathcal{B}) = \mathrm{signature}\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 8 & 0 & 32 \\ 0 & 0 & 8 & 6 & 0 & 26 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 32 & 26 & 0 & 132 \end{pmatrix} = 3$$

and that the number of real solutions of our polynomial system is equal to 3.

Another way of computing the number of real solutions of a zero-dimensional polynomial system proceeds by using the *rational univariate representation* (**Recipe VII** in Chapter 2). In this way the problem is reduced to applying **CRS**$_1$ to the minimal polynomial of the given separating element for $\mathcal{P}$.

Recipe CRS$_4$: Counting real solutions for a zero-dimensional polynomial system $\mathcal{P}$.

- ○ Compute the rational univariate representation for $\mathcal{P}$ and let u be the corresponding separating element.
- ○ Apply `Recipe CRS`$_1$ in order to get the number of real solutions of $\chi(u,T)$, which is the number of real solutions of $\mathcal{P}$.

`Recipe SQ`$_4$: Counting the Sturm Query of h for a zero-dimensional ideal $\mathcal{P}$.

- ○ Compute the rational univariate representation for $\mathcal{P}$ and let u be the corresponding separating element.
- ○ Write n' for the smallest even number bigger than n, and define
$$B(T) = g_u(1,T)^{n'} h(\frac{g_u(X_1,T)}{g_u(1,T)}, \ldots, \frac{g_u(X_k,T)}{g_u(1,T)}).$$
Apply `Recipe SQ`$_1$ to $A = \chi(u,T)$ and $B(T)$, which provides $SQ(\mathcal{P},h)$.

Example 3.4. For the Cassou-Noguès polynomial system of equations coming from the `PoSSo` test suite,

$$\begin{aligned}
P_1 = {} & 15b^4cd^2 + 6b^4c^3 + 21b^4c^2d - 144b^2c - 8b^2c^2e - 28b^2cde - 648b^2d \\
& +36b^2d^2e + 9b^4d^3 - 120 \\
P_2 = {} & 30c^3b^4d - 720db^2c - 24c^3b^2e - 432c^2b^2 + 576ec - 576de + 16cb^2d^2e \\
& -32de^2c + 16d^2e^2 + 16e^2c^2 + 9c^4b^4 + 5184 + 39d^2b^4c^2 + 18d^3b^4c \\
& -432d^2b^2 + 24d^3b^2e - 16c^2b^2de - 240c \\
P_3 = {} & 216db^2c - 162d^2b^2 - 81c^2b^2 + 5184 + 1008ec - 1008de + 15c^2b^2de \\
& -15c^3b^2e - 80de^2c + 40d^2e^2 + 40e^2c^2 \\
P_4 = {} & 261 + 4db^2c - 3d^2b^2 - 4c^2b^2 + 22ec - 22de,
\end{aligned}$$

the variable b is separating and thus the corresponding characteristic polynomial is given by

$$\begin{aligned}
\chi_b(T) = T^{16} & - \tfrac{11328065425280}{5581434681}T^{14} - \tfrac{19829599450089290240}{4068865882449}T^{12} \\
& - \tfrac{38925480508049063936}{8898609684915963}T^{10} + \tfrac{145090425457775476736}{6487086460303737027}T^8 \\
& + \tfrac{312154445605949 2499456}{1418725808868427 2878049}T^6 + \tfrac{59526120541949788 16}{344750371555027830 9365907}T^4 \\
& + \tfrac{1106372580333332224}{226190718777253759877497 15827}T^2 + \tfrac{281474976710656}{4946791019658539728520862 8513649}.
\end{aligned}$$

Applying `Recipe CRS`$_1$ to this polynomial, we find that the Cassou-Noguès polynomial system of equations has exactly four different real solutions.

4. The Sign Determination Scheme

In this section, we suppose given polynomials $h_1, \dots, h_s$ in $K[X_1, \dots, X_k]$ and $\varepsilon_i \in \{+, -, 0\}$ for $i \in \{1, \dots, s\}$. We consider the problem of determining the existence of solutions for the system

$$\operatorname{sign}(h_1(X)) = \varepsilon_1, \dots, \operatorname{sign}(h_s(X)) = \varepsilon_s \qquad (\mathcal{S}_1)$$

at the real zeros of a polynomial system $\mathcal{P}$ (see [4, 42]).

The *sign determination scheme* will solve the following problem: for $\mathcal{P}$ a polynomial system, $h_1, \dots, h_s$ polynomials in $K[X_1, \dots, X_k]$, determine which are the sign conditions to be satisfied by the polynomials $h_1, \dots, h_s$ when evaluated on the solutions of $\mathcal{P}$ in R. The case $s = 1$ is very easily solved with the help of **Recipe** $\mathtt{SQ}_1$ (when $k = 1$) or **Recipe** $\mathtt{SQ}_2$, $\mathtt{SQ}_3$ or $\mathtt{SQ}_4$ (when $k > 1$): we already know that

$$\begin{aligned} c_{[0]}(\mathcal{P}; h_1) + c_{[+]}(\mathcal{P}; h_1) + c_{[-]}(\mathcal{P}; h_1) &= SQ(\mathcal{P}, 1) \\ c_{[+]}(\mathcal{P}; h_1) - c_{[-]}(\mathcal{P}; h_1) &= SQ(\mathcal{P}, h_1). \end{aligned}$$

Also

$$c_{[+]}(\mathcal{P}; h_1) + c_{[-]}(\mathcal{P}; h_1) = SQ(\mathcal{P}, h_1^2)$$

Proposition 4.1.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{bmatrix} c_{[0]}(\mathcal{P}; h_1) \\ c_{[+]}(\mathcal{P}; h_1) \\ c_{[-]}(\mathcal{P}; h_1) \end{bmatrix} = \begin{bmatrix} SQ(\mathcal{P}, 1) \\ SQ(\mathcal{P}, h_1) \\ SQ(\mathcal{P}, h_1^2) \end{bmatrix}.$$

Thus, the computation of three Sturm Queries ($\mathcal{P}$ and 1, $\mathcal{P}$ and h_1, $\mathcal{P}$ and h_1^2) and the solving of a linear system of equations provides the integers $c_{[0]}(\mathcal{P}; h_1)$, $c_{[+]}(\mathcal{P}; h_1)$, and $c_{[-]}(\mathcal{P}; h_1)$. To continue this study, we need to generalize the definition of $c_{[+]}(\mathcal{P}; h_1)$ to a family of polynomials.

Definition 4.2. Let $\mathcal{P}$ be a zero-dimensional ideal in $K[X]$, $\mathcal{H} = (h_1, \dots, h_s)$ polynomials in $K[X]$ and $\varepsilon_1, \dots, \varepsilon_s$ a family of sign conditions (i.e., every ε_i is an element of $\{+, -, 0\}$). The integer $c_{[\varepsilon_1, \dots, \varepsilon_s]}(\mathcal{P}; \mathcal{H})$ is defined as the number of real solutions of $\mathcal{P}$ such that the sign of each h_i in such a solution is equal to ε_i:

$$c_{[\varepsilon_1, \dots, \varepsilon_s]}(\mathcal{P}; \mathcal{H}) = \#\{x \in \mathcal{Z}_R(\mathcal{P}) \mid \operatorname{sign}(h_j(x)) = \varepsilon_j, \quad 1 \le j \le s\}$$

The case $s = 2$ is solved in a similar way as before. For example, if $\mathcal{H} = (h_1, h_2)$, then

$$\begin{aligned} SQ(\mathcal{P}, h_1 h_2) &= c_{[+]}(\mathcal{P}; h_1 h_2) - c_{[-]}(\mathcal{P}; h_1 h_2) \\ &= c_{[++]}(\mathcal{P}; \mathcal{H}) + c_{[--]}(\mathcal{P}; \mathcal{H}) \\ &\quad - c_{[+-]}(\mathcal{P}; \mathcal{H}) - c_{[-+]}(\mathcal{P}; \mathcal{H}). \end{aligned}$$

Thus, the solution for $s = 2$ requires the computation of 9 Sturm Queries and the solving of the following linear system of equations of order 9.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{bmatrix} \mathrm{c}_{[00]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[+0]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[-0]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[0+]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[++]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[-+]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[0-]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[+-]}(\mathcal{P};\mathcal{H}) \\ \mathrm{c}_{[--]}(\mathcal{P};\mathcal{H}) \end{bmatrix} = \begin{bmatrix} SQ(\mathcal{P},1) \\ SQ(\mathcal{P},h_1) \\ SQ(\mathcal{P},h_1^2) \\ SQ(\mathcal{P},h_2) \\ SQ(\mathcal{P},h_1h_2) \\ SQ(\mathcal{P},h_1^2h_2) \\ SQ(\mathcal{P},h_2^2) \\ SQ(\mathcal{P},h_1h_2^2) \\ SQ(\mathcal{P},h_1^2h_2^2) \end{bmatrix} .$$

This is clearly not a good strategy since for computing the sign conditions realized by the polynomials $h_1, \ldots, h_s$ at the real solutions of $\mathcal{P}$, precisely 3^s Sturm Queries are required whereas the total number of solutions and thus of nonempty sign conditions is independent of s. To overcome this problem (see [4], [11] or [43]) we use the simple fact that the number of sign conditions looked for is bounded by the number of real solutions of $\mathcal{P}$; we accordingly reduce the linear system once solved, by removing those sign conditions that are already known to be not realized by any real solution of $\mathcal{P}$ (see [42] for complete proofs).

Recipe SI: Sign Determination Scheme.

In the initialization step, the values $SQ(\mathcal{P},1)$, $SQ(\mathcal{P},h_1)$, $SQ(\mathcal{P},h_1^2)$ are computed, and the linear system

$$A_1 \cdot C_1 = V_1 : \qquad \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{bmatrix} \mathrm{c}_{[0]}(\mathcal{P};h_1) \\ \mathrm{c}_{[+]}(\mathcal{P};h_1) \\ \mathrm{c}_{[-]}(\mathcal{P};h_1) \end{bmatrix} = \begin{bmatrix} SQ(\mathcal{P},1) \\ SQ(\mathcal{P},h_1) \\ SQ(\mathcal{P},h_1^2) \end{bmatrix}$$

is solved. The initialization step ends by the following determination of a linear system $B_1 \cdot D_1 = W_1$.

$\star_1$The columns of A_1 corresponding to the zero solutions in C_1 are removed and from this matrix a square and full rank matrix B_1 is extracted.

$\star_2$The vector D_1 is obtained from C_1 by removing those elements which are equal to 0.

$\star_3$The vector W_1 is obtained from V_1 by keeping the elements corresponding to the columns taken from A_1 to get B_1. We denote by $K_1, \ldots, K_{u_1}$ the polynomials whose Sturm queries appear in W_1.

In the $j+1$-st step, a linear system $B_j \cdot D_j = W_j$ as well as the list $K_1, \ldots, K_{u_j}$ is obtained with information on the behaviour of the real roots of $\mathcal{P}$ on the polynomials $h_1, \ldots, h_j$, and the situation for h_{j+1} is going to be analyzed. First the linear system $A_{j+1} \cdot C_{j+1} = V_{j+1}$ is constructed where

$$A_{j+1} = \begin{pmatrix} B_j & B_j & B_j \\ 0 & B_j & -B_j \\ 0 & B_j & B_j \end{pmatrix},$$

$$D_j = \begin{bmatrix} c_{[\delta_1^1 \ldots \delta_j^1]}(\mathcal{P}; h_1, \ldots, h_j) \\ \vdots \\ c_{[\delta_1^u \ldots \delta_j^u]}(\mathcal{P}; h_1, \ldots, h_j) \end{bmatrix} \Rightarrow C_{j+1} = \begin{bmatrix} c_{[\delta_1^1 \ldots \delta_j^1 0]}(\mathcal{P}; h_1, \ldots, h_j) \\ \vdots \\ c_{[\delta_1^u \ldots \delta_j^u 0]}(\mathcal{P}; h_1, \ldots, h_j) \\ c_{[\delta_1^1 \ldots \delta_j^1 +]}(\mathcal{P}; h_1, \ldots, h_j) \\ \vdots \\ c_{[\delta_1^u \ldots \delta_j^u +]}(\mathcal{P}; h_1, \ldots, h_j) \\ c_{[\delta_1^1 \ldots \delta_j^1 -]}(\mathcal{P}; h_1, \ldots, h_j) \\ \vdots \\ c_{[\delta_1^u \ldots \delta_j^u -]}(\mathcal{P}; h_1, \ldots, h_j) \end{bmatrix},$$

$$W_j = \begin{bmatrix} SQ(\mathcal{P}; K_1) \\ \vdots \\ SQ(\mathcal{P}; K_{u_j}) \end{bmatrix} \Rightarrow V_{j+1} = \begin{bmatrix} SQ(\mathcal{P}; K_1) \\ \vdots \\ SQ(\mathcal{P}; K_{u_j}) \\ SQ(\mathcal{P}; K_1 h_{j+1}) \\ \vdots \\ SQ(\mathcal{P}; K_{u_j} h_{j+1}) \\ SQ(\mathcal{P}; K_1 h_{j+1}^2) \\ \vdots \\ SQ(\mathcal{P}; K_{u_j}) h_{j+1}^2 \end{bmatrix},$$

Next, the linear system $A_{j+1} \cdot C_{j+1} = V_{j+1}$ is solved and the new system $B_{j+1} \cdot D_{j+1} = W_{j+1}$, together with $K_1, \ldots, K_{u_{j+1}}$, the output of step $j+1$, is constructed by using the same rules $(\star_1)$, $(\star_2)$, $(\star_3)$ as in the initialization step.

Example 4.3. The output of SI applied to the polynomials

$$\begin{aligned} &P = x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120, \\ &h_1 = x^3 - 2x + 1, \qquad\qquad h_2 = 3x^4 - x + 2, \end{aligned}$$

is the linear system:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \cdot \begin{bmatrix} c_{[0+]}(P; h_1, h_2) = 1 \\ c_{[++]}(P; h_1, h_2) = 4 \end{bmatrix} = \begin{bmatrix} SQ(P, 1) = 5 \\ SQ(P, h_1) = 4 \end{bmatrix}.$$

If a new polynomial, h_3, is going to be considered, then it is not necessary to start again: it is enough to continue the process with the previous system and h_3 as described in Recipe SI.

We consider now, in the univariate case, the more general problem of determining the existence of a solution for the system

$$\mathbf{sign}(\boldsymbol{h}_1(\boldsymbol{X})) = \boldsymbol{\varepsilon}_1, \ldots, \mathbf{sign}(\boldsymbol{h}_s(\boldsymbol{X})) = \boldsymbol{\varepsilon}_s, \qquad (\mathcal{S}_1)$$

where $h_1, \ldots, h_s$ are polynomials in $K[X]$ and $\varepsilon_i \in \{+,-,0\}$ for $i \in \{1, \ldots, m\}$. This problem is reduced to the case where one of the ϵ_i is equal to 0. To this end we define the polynomial P by

$$h = \prod_{i=1}^{s} h_i \,, \qquad\qquad P = (1+h^2)^2 \cdot \frac{\mathrm{d}}{\mathrm{d}X}\left(\frac{h}{1+h^2}\right) = h'(1-h^2).$$

It is very easy to verify that the system $\mathcal{S}_1$ has a solution if and only if the system

$$P(X) = 0, \mathrm{sign}(h_1(X)) = \varepsilon_1, \ldots, \mathrm{sign}(h_s(X)) = \varepsilon_s \qquad (\mathcal{S}_2)$$

has a solution.

In the multivariate case, this problem is also reduced to the sign determination of a set of functions at the zeros of a polynomial system (see [1, 42]).

5. Real Algebraic Numbers and Thom Codes

Computer algebra provides a new strategy for dealing with real algebraic numbers, which are the real roots of monic univariate polynomials with integer coefficients: the so-called *Thom codes*. Thom's Lemma, an easy result from Real Algebraic Geometry (see [5] or [11]) assures that, for a real closed field R and a given polynomial $A \in R[x]$, there are no two different real roots of A giving the same signs to the derivatives of A. This allows the introduction of the Thom code of a real algebraic number α as a list $[A; \epsilon_{d-1}, \epsilon_{d-2}, \ldots, \epsilon_1]$ where $\epsilon_i \in \{+,0,-\}$ and A is a polynomial in $\mathbb{Z}[X]$ with degree d such that $A(\alpha) = 0$ and for every i in $\{1, \ldots, d-1\}$ the sign of $A^{(i)}(\alpha)$ is equal to ϵ_i.

A first example is provided by the Thom codes of $\sqrt{2}$ and $-\sqrt{2}$:

$$\sqrt{2} \longrightarrow [X^2 - 2; +] \qquad -\sqrt{2} \longrightarrow [X^2 - 2; -]$$

It is clear that the computation of Thom codes for the real roots of a univariate polynomial is exactly the application of `Recipe SI` to A and the list of its derivatives sorted according to degree (starting with the one of smallest degree). To sort the real roots of a univariate polynomial once their Thom codes are known, is an easy task according to the following proposition.

Proposition 5.1. *Let A be a polynomial in $R[X]$ of degree d with α and β real roots of A with Thom codes*

$$\alpha = [A; \epsilon_{d-1}, \ldots, \epsilon_1], \qquad \beta = [A; \delta_{d-1}, \ldots, \delta_1].$$

Then:

1. *If, for $j \in \{1, \ldots, d-1\}$, $\epsilon_j = \delta_j$, then $\alpha = \beta$.*
2. *Otherwise, let k be the biggest index such that $\epsilon_k \neq \delta_k$. Then:*
   a) If $\epsilon_{k+1} = +$, then $\alpha > \beta$ if and only if $\epsilon_k > \delta_k$
   b) If $\epsilon_{k+1} = -$, then $\alpha > \beta$ if and only if $\epsilon_k < \delta_k$

There exist algorithms manipulating real algebraic numbers knowing only their Thom codes: field operations, computations in towers of fields, etc. All of these algorithms are purely symbolic and consist of several Sturm Queries computations and linear systems of equations solving, and do not require any kind of approximation of the real roots of the considered polynomial (see [11], [43] and [12]). Moreover, it is worth remarking that these algorithms do not require the squarefree hypothesis often used in algorithms computing isolating intervals for the real roots of a univariate polynomial.

Example 5.2. Let A be the polynomial

$$A = X^5 - 15X^4 + 85X^3 - 225X^2 + 274X - 120.$$

In order to compute Thom codes for the real roots of A we apply **`Recipe SI`** to A and its derivatives, starting with the one of smallest degree, $A^{(4)}$. In order to avoid unnecessary computations the process is stopped once all the real roots are characterized, i.e., when all the elements in vector C_j are equal to 1. In this example this happens after considering $A^{(4)}$, $A^{(3)}$, and $A^{(2)}$:

$$\mathtt{SI}(A, \{A^{(4)}, A^{(3)}, A^{(2)}\})$$

gives

$$SQ(A, K_1) = 5, SQ(A, K_2) = 0, SQ(A, K_3) = 4,$$
$$SQ(A, K_4) = 0, SQ(A, K_5) = 0$$

and

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{bmatrix} \mathrm{c}_{[0-0]}(A; A^{(4)}, A^{(3)}, A^{(2)}) \\ \mathrm{c}_{[+++]}(A; A^{(4)}, A^{(3)}, A^{(2)}) \\ \mathrm{c}_{[-++]}(A; A^{(4)}, A^{(3)}, A^{(2)}) \\ \mathrm{c}_{[++-]}(A; A^{(4)}, A^{(3)}, A^{(2)}) \\ \mathrm{c}_{[-+-]}(A; A^{(4)}, A^{(3)}, A^{(2)}) \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}.$$

with

$$K_1 = 1, \qquad K_2 = X - 3, \qquad K_3 = X^2 - 6X + 9,$$
$$K_4 = 2X^3 - 18X^2 + 51X - 45,$$
$$K_5 = 2X^4 - 24X^3 + 105X^2 - 198X + 135,$$

so that

$$\mathrm{c}_{[0-0]}(A; A^{(4)}, A^{(3)}, A^{(2)}) = 1$$
$$\mathrm{c}_{[+++]}(A; A^{(4)}, A^{(3)}, A^{(2)}) = 1$$
$$\mathrm{c}_{[-++]}(A; A^{(4)}, A^{(3)}, A^{(2)}) = 1$$
$$\mathrm{c}_{[++-]}(A; A^{(4)}, A^{(3)}, A^{(2)}) = 1$$
$$\mathrm{c}_{[-+-]}(A; A^{(4)}, A^{(3)}, A^{(2)}) = 1.$$

Thus, Thom codes for the real roots of A are

$$\boldsymbol{\alpha_1 = [0 - 0], \qquad \alpha_2 = [+ + +], \qquad \alpha_3 = [- + +],}$$

$$\alpha_4 = [++-], \qquad \alpha_5 = [-+-].$$

Applying Proposition 5.1, we obtain the sorting of the α_i:

$$\alpha_5 < \alpha_3 < \alpha_1 < \alpha_4 < \alpha_2$$

Finally, to determine the sign of $\alpha_3^4 - \alpha_3 - 1$, it is enough to apply again Recipe SI to the linear system shown before with the polynomial $h = X^4 - X - 1$. In fact, this gives the sign behaviour of h on the α_i:

$$h(\alpha_1) > 0, \quad h(\alpha_2) > 0, \quad h(\alpha_3) > 0, \quad h(\alpha_4) > 0, \quad h(\alpha_5) < 0.$$

Practical experience has shown that, in general, manipulation of real algebraic numbers by means of Thom codes is less efficient than the use of isolating intervals. However, Thom codes have the following advantages: first, they are more stable since once they have been computed they do not need to be refined and, second, they are the only way of dealing with real algebraic numbers over non-Archimedean ordered fields. This last statement may be clarified by the following example.

Example 5.3. Non-Archimedean ordered fields appear quite frequently as very useful tools when considering algorithms dealing with topological questions of real algebraic sets (for instance plane curves, see [12]) or quantifier elimination methods with low complexity, see [26]).

When studying real algebraic plane curves, non-Archimedean ordered fields can be used to decide the topological type of the considered curve around a singularity. If we consider the real algebraic curve C defined by the polynomial

$$A = 2X^4 - 3X^2Y + Y^4 - 2Y^3 + Y^2,$$

then it is well known that the X-coordinates of the singular points are contained in the set of real roots of the discriminant D of A with respect to the variable Y:

$$D = X^6(2048X^6 - 4608X^4 + 37X^2 + 12).$$

The polynomial D has 5 real roots; let α be the smallest one, characterized by its Thom code. To compute the behaviour of the curve C to the right of the vertical line $X = \alpha$, the polynomial A is considered as a polynomial in the variable Y with coefficients in $\mathbb{Q}(X)$:

$$A = Y^4 - 2Y^3 + Y^2 - (3X^2)Y + 2X^4.$$

Next the field $\mathbb{Q}(X)$ is ordered by saying that X is bigger than α and smaller than every rational number bigger than α. In this context the number of real roots of A (Y being the unknown) in the real closure of $\mathbb{Q}(X)$ is equal to the number of branches of C that touch to the right of the vertical line $X = \alpha$. In this case the number of real roots can be computed by determining the principal Sylvester-Habicht coefficients of A and A', considered as polynomials in Y, $\{h_k(X)\}_{k=0,\dots,4}$; applying Recipe CRS$_1$, we find that the number is

2. The conclusion is that there are two branches of C leaving from the right of the vertical line $X = \alpha$. Note that, to determine the sign of an element $q(X) \in \mathbb{Q}[X]$, it is sufficient to determine the sign of the first non-vanishing derivative of $q(X)$ when evaluated in α: so it is enough to apply `Recipe SI`.

The field $\mathbb{Q}(X)$, ordered as described before, is non-Archimedean since, by definition, there are no rational numbers between α and X.

6. Quantifier Elimination

One of the main problems in Computational Real Algebraic Geometry is the development of efficient Quantifier Elimination algorithms. Tarski's Theorem assures that, for any quantified formula on sign conditions of polynomials, it is possible to construct algorithmically a quantifier free formula equivalent to the initial one. For example,

$$\exists\, x \in \mathbb{R} \quad x^2 + bx + c = 0 \qquad \Longleftrightarrow \qquad b^2 - 4c \geq 0.$$

As this problem, in its more general form, is well known to be unsolvable in polynomial time (see [13]), one way of attacking this problem is the isolation of specific and particular cases where efficient algorithms can be applied. In this context, the word efficient does not mean polynomial time: we search algorithms, methods, and criterions allowing us to perform Quantifier Elimination on formulae with a fixed structure and low degrees of the involved polynomials (see for example [28], [27] or [47]).

In this section, some recipes are presented according to this methodology: easy to describe methods performing Quantifier Elimination on formulae with a prescribed structure. The first case to be considered is possibly the easiest one: one quantifier and one equation or inequality (following [21]).

Proposition 6.1. *Let d be a positive even integer and*

$$A_d(\underline{a}, X) = X^d + a_{d-1}X^{d-1} + \ldots + a_1 X + a_0.$$

Put

$$\begin{array}{rll} \mathbf{H}_d: & \forall\, x & A_d(\underline{a}, x) > 0, \\ \mathrm{e}_d: & \exists\, x & A_d(\underline{a}, x) = 0, \\ \Lambda_d: & \exists\, x & A_d(\underline{a}, x) < 0. \end{array}$$

Then:

$$\begin{array}{rcl} \mathbf{H}_d & \Longleftrightarrow & D([\mathrm{syha}_d(A_d, A'_d), \ldots, \mathrm{syha}_0(A_d, A'_d)]) = 0, \\ \mathrm{e}_d & \Longleftrightarrow & D([\mathrm{syha}_d(A_d, A'_d), \ldots, \mathrm{syha}_0(A_d, A'_d)]) > 0, \\ \Lambda_d & \Longleftrightarrow & \displaystyle\bigvee_{j=1}^{d/2-1} D([\{\mathrm{syha}_k(R_{d,j}, R'_{d,j}({A_d^{(2j-1)}})^{2})\}_{0 \leq k \leq 2d}]) > 0, \end{array}$$

with

$$R_{d,j} = \sum_{k=0}^{2j-2} (A_d^{(k)}(\underline{a}, x))^2.$$

Conditions equivalent to $\mathbf{H}_d$ and e_d are presented as a union of basic semi-algebraic sets obtained by regarding all the 3^{d-2} possible sign conditions over the polynomials $\mathrm{syha}_i(A_d, A'_d)$ and keeping those making 0 (for $\mathbf{H}_d$) or > 0 (for e_d) the function D. The situation for Λ_d is similar but a little more complicated.

Example 6.2. For the first nontrivial case, $d = 4$, the formula $\mathbf{H}_4$ is equivalent to the union of the following 9 semi-algebraic basic sets:

$$\begin{array}{cccccccc} & \bigl[>0,<0,>0\bigr] & \cup & \bigl[<0,>0,>0\bigr] & \cup & \bigl[<0,<0,>0\bigr] & \cup \\ \cup & \bigl[<0,=0,>0\bigr] & \cup & \bigl[<0,=0,<0\bigr] & \cup & \bigl[=0,>0,<0\bigr] & \cup \\ \cup & \bigl[=0,<0,>0\bigr] & \cup & \bigl[=0,=0,>0\bigr] & \cup & \bigl[<0,=0,=0\bigr], \end{array}$$

with $[\tau_2, \tau_1, \tau_0]$ denoting the semi-algebraic set defined by $S_2 \tau_2 0, S_1 \tau_1 0, S_0 \tau_0 0$, and where the polynomial S_i is defined as follows.

$$\begin{aligned} S_2 = {} & 3a_3^2 - 8a_2 \\ S_1 = {} & 2a_2^2a_3^2 - 8a_2^3 + 32a_2a_0 + a_1a_2a_3 - 12a_3^2a_0 - 6a_1a_3^3 - 36a_1^2 \\ S_0 = {} & -27a_1^4 - 4a_3^3a_1^3 + 18a_2a_3a_1^3 - 6a_3^2a_0a_1^2 + 144a_2a_0a_1^2 + a_2^2a_3^2a_1^2 \\ & -4a_2^3a_1^2 - 192a_3a_0^2a_1 + 18a_0a_2a_3^3a_1 - 80a_0a_2^2a_3a_1 + 256a_0^3 \\ & -27a_3^4a_0^2 + 144a_2a_3^2a_0^2 - 128a_2^2a_0^2 - 4a_2^3a_3^2a_0 + 16a_2^4a_0. \end{aligned}$$

The previous description for $\mathbf{H}_4$ can easily be reduced to the following one.

$$\begin{aligned} & \bigl[<0,\neq 0,>0\bigr] \cup \bigl[=0,\leq 0,>0\bigr] \cup \bigl[>0,<0,>0\bigr] \cup \bigl[=0,>0,<0\bigr] \cup \\ & \cup \bigl[S_2<0, S_1=0\bigr] = \end{aligned}$$

$$\bigl[<0,\neq 0,>0\bigr] \cup \bigl[=0,\leq 0,>0\bigr] \cup \bigl[>0,<0,>0\bigr] \cup \bigl[S_2<0, S_1=0\bigr].$$

The last simplification is due to the following fact:

$$S_2 = 0 \quad \Longrightarrow \quad a_2 = \frac{3a_3^2}{8} \quad \Longrightarrow \quad S_1 = -(16a_1 + a_3^3)^2 \leq 0.$$

For e_4 we get the union of the following 16 semi-algebraic basic sets:

$$\begin{array}{l} \bigl[>0,>0,>0\bigr] \cup \bigl[>0,>0,<0\bigr] \cup \bigl[>0,<0,<0\bigr] \cup \bigl[<0,<0,<0\bigr] \cup \\ \bigl[>0,=0,>0\bigr] \cup \bigl[>0,=0,<0\bigr] \cup \bigl[=0,>0,>0\bigr] \cup \bigl[=0,<0,<0\bigr] \cup \\ \bigl[=0,=0,<0\bigr] \cup \bigl[>0,>0,=0\bigr] \cup \bigl[>0,<0,=0\bigr] \cup \bigl[<0,<0,=0\bigr] \cup \\ \bigl[=0,>0,=0\bigr] \cup \bigl[=0,<0,=0\bigr] \cup \bigl[>0,=0,=0\bigr] \cup \bigl[=0,=0,=0\bigr]. \end{array}$$

The same strategy as for $\mathbf{H}_4$ allows us to obtain a simplified description:

$$\mathrm{e}_4 = \bigl[S_2 = 0, S_0 \leq 0\bigr] \cup \bigl[>0,<0,\leq 0\bigr] \cup \bigl[<0,<0,\leq 0\bigr] \cup \bigl[S_2 > 0, S_1 \geq 0\bigr].$$

Finally, we remark that a set appearing in this description for $\mathbf{H}_4$ or e_4 may be empty.

One natural generalization of the previous case is the following Quantifier Elimination problem:

$$\exists\, x_1 \in \mathbb{R}, \ldots, \exists\, x_k \in \mathbb{R} \qquad \begin{cases} A_1 = x_1^{d_1} + Q_1(\underline{t}, x_1, \ldots, x_k) & = 0 \\ A_2 = x_2^{d_2} + Q_2(\underline{t}, x_1, \ldots, x_k) & = 0 \\ \qquad\qquad \vdots & \ \vdots \\ A_k = x_k^{d_k} + Q_k(\underline{t}, x_1, \ldots, x_k) & = 0, \end{cases}$$

where every $Q_i(\underline{t}, X_1, \ldots, X_k)$ is a polynomial in $\mathbb{Z}[\underline{t}, X_1, \ldots, X_k]$ with total degree (in the X's) smaller than d_i and $\underline{t} = (t_1, \ldots, t_k)$. This kind of polynomial system is usually called a *Pham System*. The restriction on the structure (and number) of the polynomials in the system is due to the need of controlling the Gröbner basis of $A_1, \ldots, A_k$. In this situation, for any specialization of the parameters $\underline{t}$, the set $\{A_1, \ldots, A_k\}$ is a Gröbner basis with respect to the total degree ordering, and the monomials $X_1^{\alpha_1} \cdots X_k^{\alpha_k}$ $(0 \le \alpha_i < d_i, 1 \le i \le n)$ give the basis $\mathcal{A}$ of §3. This allows us to construct the *trace matrix* Tr whose entries will be polynomials in the parameters $\underline{t}$ and thus:

$$\exists\, x_1 \in \mathbb{R} \cdots \exists\, x_k \in \mathbb{R}\ A_1(x) = 0 \wedge \ldots \wedge A_k(x) = 0 \iff \text{signature}(\mathrm{Tr}) > 0.$$

The problem has been reduced to parameterizing, or finding a closed expression, for the signature of a symmetric matrix depending polynomially on several parameters. This can be accomplished using Proposition 3.2.

Example 6.3. For the Pham System

$$\begin{aligned} A_1 &= X_1^2 + u_1 X_1 + v_1 X_2 + w_1, \\ A_2 &= X_2^2 + u_2 X_1 + v_2 X_2 + w_2, \end{aligned}$$

the monomial set $\mathcal{A}$ is

$$\mathcal{A} = \{1, X_1, X_2, X_1 X_2\}$$

and the trace matrix has the following form

$$\mathrm{Tr} = \begin{pmatrix} S_{00} & S_{10} & S_{01} & S_{11} \\ S_{10} & S_{20} & S_{11} & S_{21} \\ S_{01} & S_{11} & S_{02} & S_{12} \\ S_{11} & S_{21} & S_{12} & S_{22} \end{pmatrix},$$

with

$$\begin{aligned} S_{00} &= 4 \\ S_{01} &= -2v_2 \\ S_{10} &= -2u_1 \\ S_{20} &= 2u_1^2 + 2v_1 v_2 - 4w_1 \\ S_{11} &= 3v_1 u_2 + v_2 u_1 \\ S_{02} &= 2u_2 u_1 + 2v_2^2 - 4w_2 \\ S_{12} &= -2u_2 u_1^2 - 5u_2 v_1 v_2 + 4u_2 w_1 - v_2^2 u_1 + 2w_2 u_1 \\ S_{21} &= -5u_1 v_1 u_2 - v_2 u_1^2 - 2v_1 v_2^2 + 4v_1 w_2 + 2v_2 w_1 \\ S_{22} &= 8u_2 u_1 v_1 v_2 + u_1^2 v_2^2 - 2u_1^2 w_2 + 3v_1^2 u_2^2 - 6v_1 v_2 w_2 \\ &\quad -6u_2 u_1 w_1 - 2w_1 v_2^2 + 4w_1 w_2 + 2u_2 u_1^3 + 2v_1 v_2^3. \end{aligned}$$

The signature of this symmetric matrix determines the solution of any Quantifier Elimination on the polynomial system of equations

$$A_1(x_1, x_2) = 0, \quad A_2(x_1, x_2) = 0$$

via its characteristic polynomial.

This approach tends to generate very complicated expressions when computing the characteristic polynomial (see the example below). A way of avoiding this problem was proposed in [22]; it involves changing the set $\mathcal{A}$, in such a way that Tr obtains a Hankel structure, resulting in a much easier signature parameterization.

Example 6.4. For the Pham System

$$A = X^2 + bX + c,$$

the monomial set $\mathcal{A}$ is

$$\mathcal{A} = \{1, X\}$$

and the trace matrix has the following form.

$$\mathrm{Tr} = \begin{pmatrix} S_{00} & S_{10} \\ S_{10} & S_{20} \end{pmatrix} = \begin{pmatrix} 2 & -b \\ -b & b^2 - 2c \end{pmatrix}$$

The use of Proposition 3.2 for the characteristic polynomial of Tr,

$$\lambda^2 + (2c - b^2 - 2)\lambda + (b^2 - 4c),$$

provides the following characterization for the different values of the signature for S_1.

$$\mathrm{signature}(\mathrm{Tr}) = 2 \iff 2c - b^2 - 2 < 0, b^2 - 4c > 0$$
$$\mathrm{signature}(\mathrm{Tr}) = 1 \iff 2c - b^2 - 2 < 0, b^2 - 4c = 0$$
$$\mathrm{signature}(\mathrm{Tr}) = 0 \iff \begin{cases} 2c - b^2 - 2 = 0,\ b^2 - 4c > 0 \text{ or} \\ 2c - b^2 - 2 > 0,\ b^2 - 4c < 0 \text{ or} \\ 2c - b^2 - 2 < 0,\ b^2 - 4c < 0 \text{ or} \\ 2c - b^2 - 2 = 0,\ b^2 - 4c < 0 \text{ or} \\ 2c - b^2 - 2 = 0,\ b^2 - 4c = 0. \end{cases}$$

Two cases have been removed since they provided negative signature. Here it is easy to work out the above formulae and find the simpler result, but in general this is a very complicated problem. Fortunately, the matrix Tr is Hankel and its signature is uniquely determined by the sign of its principal minors: 2 and $b^2 - 4c$. In this particular case, the signature of Tr is equal to the difference of the number of sign variations and the number of sign permanences in $\{1, 2, b^2 - 4c\}$. Thus, the classical result is obtained automatically:

$$\mathrm{signature}(S_1) = \begin{cases} 2 & \text{if } b^2 - 4c > 0 \\ 1 & \text{if } b^2 - 4c = 0 \\ 0 & \text{if } b^2 - 4c < 0. \end{cases}$$

The extension of this Hankel approach to a general Pham System can be found in [22].

We end this section by showing how to test the emptiness of a hypersurface in $\mathbb{R}^k$, i.e., how to perform Quantifier Elimination on the formulae

$$\exists\, x_1 \in \mathbb{R}, \ldots, \exists\, x_k \in \mathbb{R} \qquad P(x_1, \ldots, x_k) = 0,$$

where P is a degree d polynomial in $\mathbb{Z}[X_1, \ldots, X_k]$. First, let X_{k+1} and Ω be new variables such that $\mathbb{R}[\Omega]$ is ordered by assuming that $1/\Omega$ is infinitesimal: positive and smaller than every positive rational number. Next, the polynomial P is deformed in the following way

$$P \longmapsto Q = P^2 + \left(\sum_{i=1}^{k+1} X_i^2 - \Omega^2\right)^2$$

with $Q \in \mathbb{Z}[\Omega][X_1, \ldots, X_k]$. Next, a new variable ζ is introduced such that ζ is infinitesimal with respect $\mathbb{R}[\Omega]$ and a new deformation is performed:

$$Q \mapsto R = (1-\zeta)Q + \zeta\left(\sum_{i=1}^{k} X_i^{2(d+1)} + X_{k+1}^6 - (k+1)(\Omega+1)^{2(d+1)}\right).$$

According to [42], the hypersurface $R(X_1, \ldots, X_{k+1}) = 0$ in L^{k+1} is smooth and bounded by $\Omega+1$, where L denotes the real closure of $\mathbb{R}(\Omega, \zeta)$. Moreover, the set S of critical points with respect to X_1 of this hypersurface

$$\begin{aligned} &R([\Omega,\zeta]; x_1, \ldots, x_{k+1}) = 0 \\ &\frac{\partial R}{\partial X_2}([\Omega,\zeta]; X_1, \ldots, X_{k+1}) = 0 \\ &\qquad\qquad\vdots \\ &\frac{\partial R}{\partial X_{k+1}}([\Omega,\zeta]; X_1, \ldots, X_{k+1}) = 0 \end{aligned}$$

is finite since the previous polynomial system is a Pham System. Their solution in L^{k+1} and the consideration of $\zeta \to 0$ provides the answer: if there exists a real solution in L^{k+1} such that the limit for $\zeta \to 0$ exists, then our initial hypersurface is nonempty.

7. Appendix: Properties of the Polynomials in the Sylvester-Habicht Sequence

This appendix is concerned with technical details and proofs of some results in §2.

7.1 Definition and the Structure Theorem

Polynomials in the Sylvester-Habicht sequence of (A, B) are polynomials which are proportional to polynomials in the remainder sequence of (A, B) but have better properties such as bit size control and good behaviour under specialization. They are defined through determinants and obtained by a sign modification from the subresultant sequence of (A, B). For an integer $d \geq 1$, we call (A, B) a *(regular) d-couple* if $d = \deg A > \deg B$. If d is clear from the context, we speak simply of a *regular couple.*

Remark 7.1. We shall assume $d = \deg A > \deg B$ throughout this section. If (A, B) is not a regular d-couple, we can replace B by $B_1 = \mathrm{Rem}(a_d^{2e}B, A)$, where $2e$ is the smallest even number $\geq q - d + 1$. This does not modify the Cauchy index. In general, we define the sequence $H_j(A, B)$, for $j = 0, \ldots, d$, by $H_j(A, B_1)$ if $q = \deg(B) \geq d$.

Let D be a domain, K its quotient field. For a regular d-couple (A, B) in $D[X]$ and $j \leq d - 1$, let the j-th *Sylvester matrix* of (A, B), denoted by $\mathrm{syl}_j = \mathrm{syl}_j(A, B)$, be the matrix whose rows are the coefficient vectors of the polynomials

$$AX^{d-2-j}, AX^{d-3-j}, \ldots, AX, A, B, BX, \ldots, BX^{d-2-j}, BX^{d-1-j}$$

with respect to the monomial basis

$$X^{2d-2-j}, X^{2d-3-j}, \ldots, X, 1.$$

This matrix has $2d - 1 - 2j = (d - 1 - j) + (d - j)$ rows and $2d - 1 - j$ columns. If

$$\begin{array}{rcll} A & = & a_dX^d + & a_{d-1}X^{d-1} + a_{d-2}X^{d-2} + \cdots + a_0, \\ B & = & & b_{d-1}X^{d-1} + b_{d-2}X^{d-2} + \cdots + b_0 \end{array}$$

(allowing top terms of B to vanish), then syl_j has the shape

$$\mathrm{syl}_j = \underbrace{\left(\begin{array}{cccccccc} a_d & \cdots & \cdots & \cdots & \cdots & a_0 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & a_d & \cdots & \cdots & \cdots & \cdots & a_0 \\ & & & b_{d-1} & \cdots & \cdots & \cdots & b_0 \\ & & b_{d-1} & \cdots & \cdots & \cdots & b_0 & \\ & \iddots & \iddots & \iddots & \iddots & \iddots & & \\ b_{d-1} & \cdots & b_{j+1} & \cdots & b_0 & & & \end{array}\right)}_{2d-1-j} \begin{array}{l} \left.\vphantom{\begin{array}{c} a \\ a \\ a \end{array}}\right\} d-1-j \\ \left.\vphantom{\begin{array}{c} a \\ a \\ a \\ a \end{array}}\right\} d-j \end{array}$$

and is a submatrix of the full Sylvester matrix $\mathrm{syl}_0 = \mathrm{syl}_0(A, B)$.

Remark 7.2. We name the rows by the corresponding polynomials AX^r (respectively, BX^s) and the columns by the corresponding monomials X^t. For instance, the first column is the X^{2d-2-j}-column.

For $\ell = 0, \ldots, 2d-2-j$, let $\mathrm{syl}_{j,\ell} = \mathrm{syl}_{j,\ell}(A, B)$ be the square matrix of dimension $2d-1-2j$ obtained by taking the first $2d-2-2j$ columns of syl_j (the ones indexed by $X^{2d-2-j}, X^{2d-3-j}, \ldots, X^{j+1}$) and the X^ℓ-column of syl_j.

The *Sylvester-Habicht sequence* of a d-couple (A, B) is the sequence

$$H_d = H_d(A, B), \ldots, H_0 = H_0(A, B)$$

defined as follows.

- ○ $H_d = A$,
- ○ $H_j = \sum_{\ell=0}^{j} \det(\mathrm{syl}_{j,\ell}) X^\ell \quad \text{if } 0 \le j \le d-1.$

Notice that $H_{d-1} = B$. (Formally, we add the definitions $H_{-1} = 0$, and $\deg(0) = -1$, $\mathrm{lc}(0) = 0$ for the degree and leading coefficient of the zero polynomial.) The *sequence of principal Sylvester-Habicht coefficients*

$$h_d = h_d(A, B), \ldots, h_0 = h_0(A, B),$$

is defined as $h_j = \mathrm{coeff}_j(H_j)$, the formal leading coefficient of H_j for $0 \le j < d$ (with the extension $h_{-1} = 0$). If $h_j = 0$, the polynomial H_j is called *defective*. So $h_j \neq 0$ boils down to saying that (H_j, H_{j-1}) is a regular j-couple.

Remark 7.3.

1. For reasons of signs we consider the Sylvester-Habicht polynomials (cf. [18], [19], [20]) rather than the usual subresultants. The corresponding factor of proportionality $(-1)^{(d-j)(d-j-1)/2}$ is accomplished by the above permutation of the BX^s-rows in the definition of the j-th Sylvester matrix. So for a d-couple (A, B), the Sylvester-Habicht sequence results from the subresultant sequence by multiplying the two starting subresultants A and B by $+1$, the next two by -1 (no matter whether non-defective, defective, or vanishing), and so on. Furthermore, this permutation of the rows has organizational advantages when syl_{j-1} is considered as submatrix of syl_j; one only has to add a first and a last row in order to get syl_j.
2. For reasons of uniformity in the degree of B (and for simpler recursions), we consider throughout the Sylvester-Habicht polynomials with respect to a degree pattern $(d, d-1)$, even if $\deg(B) < d-1$.
3. The initializing definition $h_d = a_d$ differs from the tradition but appears more natural to us.

Remark 7.4. Let $\Sigma_j = \Sigma_j(A, B)$ be the $(2d-1-2j) \times (2d-2-2j)$ east block of the matrix syl_j on the columns of $X^{2d-2-j}, X^{2d-3-j}, \ldots, X^{j+2}, X^{j+1}$. The matrix Σ_j defines a linear form $\sigma_j = \sigma_j(A, B)$ given by $\sigma_j(\xi) = \det(\Sigma_j, \xi)$ for $\xi \in D^{2d-1-2j}$, which is orthogonal to these columns. This linear form σ_j is nonzero if and only if the rank of Σ_j takes its maximal value $2d-2-2j$; in this case its extension to $K^{2d-1-2j}$ is uniquely determined up to a nonzero factor in K by this orthogonality property. The coefficients $\operatorname{coeff}_\nu(H_j)$ of H_j are the values of σ_j on the X^ν-column. Therefore,

$$\begin{aligned}
H_j &= \sum_{\ell=0}^{j} \det(\mathrm{syl}_{j,\ell}) X^\ell = \sum_{\ell=0}^{2d-2-j} \det(\mathrm{syl}_{j,\ell}) X^\ell \\
&= \sum_{r=0}^{d-2-j} u_{j,r} \cdot AX^r + \sum_{t=0}^{d-1-j} v_{j,t} \cdot BX^t \\
&= \left(\sum_{r=0}^{d-2-j} u_{j,r} X^r \right) \cdot A + \left(\sum_{t=0}^{d-1-j} v_{j,t} X^t \right) \cdot B \\
&= U_j \cdot A + V_j \cdot B \in (A,B)D[X],
\end{aligned}$$

where the coefficients $u_{j,r} = u_{j,r}(A,B)$ and $v_{j,t} = v_{j,t}(A,B)$ of the polynomials $U_j = U_j(A,B)$, $V_j = V_j(A,B) \in D[X]$ of the extended gcd like expression for H_j at the right hand side (also known as a *Bézout relation*) are, up to sign, the maximal minors of Σ_j (that is, the coefficients of the linear form σ_j).

Now we are going to make the relation between the Sylvester-Habicht polynomials and remainders more precise. The main property of the Sylvester-Habicht polynomials is the following Structure Theorem, which is a refinement of the famous Subresultant Theorem (cf. [6, 7, 8, 15, 18, 25, 29, 33, 38]).

Theorem 7.5 (Structure Theorem). *For a d-couple (A,B) of polynomials in $D[X]$, D a domain with quotient field K, the polynomials in the Sylvester-Habicht sequence $H_d, \dots, H_0$ are either K-proportional to the polynomials in the signed remainder sequence $Sy^0, \dots, Sy^w$, or zero. Putting $j = \deg Sy^i$ and $k = \deg Sy^{i+1}$ for $i \leq w$, one has the K-proportionalities*

$$\begin{aligned}
Sy^i &\sim_K H_j, \\
Sy^{i+1} &\sim_K H_{j-1} \sim_K H_k.
\end{aligned}$$

Furthermore, writing $c_{j-1} = \operatorname{coeff}_k(H_{j-1})$ for the leading coefficient of H_{j-1}, the following relations hold.

1. *If $j = d$, then*

$$h_k = (-1)^{(j-k)(j-k-1)/2} \cdot c_{j-1} \cdot (c_{j-1} h_j)^{j-k-1}.$$

If $j < d$, then

$$h_k = (-1)^{(j-k)(j-k-1)/2} \cdot c_{j-1} \cdot \left(\frac{c_{j-1}}{h_j} \right)^{j-k-1}.$$

2. $H_\ell = 0$ for $k < \ell < j-1$.

3. a) Let $k \geq 0$. If $j = d$, then

$$H_{k-1} = -\mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1});$$

if $j < d$, then

$$h_j^2 \cdot H_{k-1} = -\mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1}).$$

b) These are exact signed Euclidean divisions in $D[X]$.

Remark 7.6.

1. The values of j with $h_j \neq 0$ are precisely the degrees of the polynomials in the signed Euclidean remainder sequence. The vanishing of the intermediate Sylvester-Habicht polynomials is the famous gap structure described in §2.3.1.
2. When $j < d$ and H_{j-1} is non-defective, that is, when $k = j-1$, then $h_{j-1} = c_{j-1} = h_k$ and the exact signed Euclidean division in 3 becomes Habicht's generic division formula $h_j^2 \cdot H_{j-2} = -\mathrm{Rem}(h_{j-1}^2 \cdot H_j, H_{j-1})$ (cf. [25]).
3. When H_{j-1} is defective, the preceding result is a strict improvement of the classical Subresultant Theorem.
4. Defining $\overline{h}_d = h_d^{-1}, \overline{h}_j = h_j$, as in Algorithm **SyHa**, the formulae in the Structure Theorem can be unified for $0 \leq j \leq d$ by
   a)
   $$h_k = (-1)^{(j-k)(j-k-1)/2} \cdot c_{j-1} \cdot \left(\frac{c_{j-1}}{\overline{h}_j}\right)^{j-k-1},$$
   b) $H_\ell = 0$ for $k < \ell < j-1$,
   c) for $k \geq 0$,
   $$h_j \overline{h}_j \cdot H_{k-1} = -\mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1}).$$

As an immediate consequence of the Structure Theorem, and of the remark above, we get the correctness of Algorithm **SyHa**.

Corollary 7.7. *Algorithm* **SyHa** *computes the Sylvester-Habicht sequence of A and B.*

Corollary 7.8. *Algorithm* **SyHa** *outputs polynomials in $D[X]$. When executed as indicated in Remark 2.15, all its intermediate computations take place in $D[X]$. If A and B are polynomials in $\mathbb{Z}[X]$ with coefficients of bit size τ (with $\tau > \log(p+1)$), the output and intermediate computations of the algorithm are polynomials with integer coefficients of bit size at most $O(p\tau)$.*

Proof. This is an immediate consequence of Hadamard's inequality on the size of determinants, see §3 of Chapter 3 or [34].

7.2 Proof of the Structure Theorem

Apart from Statement 3b) on the exact signed Euclidean division, which we shall prove at the end of this section, the proof of the Structure Theorem 7.5 will follow from the subsequent two lemmas. For technical reasons we modify the remainder of the Euclidean division

$$A = \mathrm{Quo}(A,B)\cdot B + \mathrm{Rem}(A,B)$$

of a couple (A,B) of nonzero polynomials in $K[X]$ with $\deg A \geq \deg B$ by multiplying this equation by $-\frac{\mathrm{lc}B}{\mathrm{lc}A}$. The *Gaussian remainder* $\mathrm{Gau}(A,B)$ is defined as

$$\begin{aligned}\mathrm{Gau}(A,B) &= \left(-\frac{\mathrm{lc}B}{\mathrm{lc}A}\right)\cdot \mathrm{Rem}(A,B)\\ &= \left(-\frac{\mathrm{lc}B}{\mathrm{lc}A}\cdot A\right) + \left(\frac{\mathrm{lc}B}{\mathrm{lc}A}\cdot \mathrm{Quo}(A,B)\right)\cdot B.\end{aligned}$$

Note that

$$\mathrm{Gau}(aA,bB) = b\cdot \mathrm{Gau}(A,B) \text{ for nonzero } a,b\in K \tag{7.1}$$

(while $\mathrm{Rem}(aA,bB) = a\cdot\mathrm{Rem}(A,B)$ for nonzero $a,b\in K$) and that the cofactor of B in the above Bézout relation for the Gaussian remainder is monic. The latter property is characteristic of the *Gaussian remainder sequence* $G_0,\ldots,G_w$, which is recursively defined by $G_{i+1} = G_{i+1}(A,B) = \mathrm{Gau}(G_{i-1},G_i)$ with starting conditions $G_0 = A$ and $G_1 = B$.

Lemma 7.9. *Let (A,B) be a d-couple of nonzero polynomials in $K[X]$ with Gaussian remainder sequence $G_0,\ldots,G_w$. Then there is a polynomial $A_i\in K[X]$ of degree d_1-d_i and a monic polynomial $B_i\in K[X]$ of degree d_0-d_i (where $d_i = \deg R_i = \deg G_i$) such that*

$$G_{i+1} = A_i\cdot A + B_i\cdot B$$

for $i = 0,\ldots,w-1$.

Proof. Induction on i.

The monic cofactor of B in the above Bézout relation for G_{i+1} allows us to perform a unimodular row manipulation (with determinant one) in the matrix syl_ℓ which replaces some of the BX^s-rows by certain $G_{i+1}X^t$-rows. Since the maximal minors of syl_ℓ remain unchanged, we can conveniently analyze the coefficients of the Sylvester-Habicht polynomials in the gap situation and bridge the gap.

We define $\overline{h}_d = h_d^{-1}$ and for $j = d_i, k = d_{i+1}$ two consecutive degrees of polynomials in the Sylvester sequence

$$\overline{h}_\ell = (-1)^{j-\ell-1} \cdot c_{j-1} \cdot \left(\frac{c_{j-1}}{\overline{h}_j}\right)^{j-\ell-1}$$

for $k \le \ell < j$. Note that, in the usual case, for $j < d$, $\overline{h}_j = h_j$.

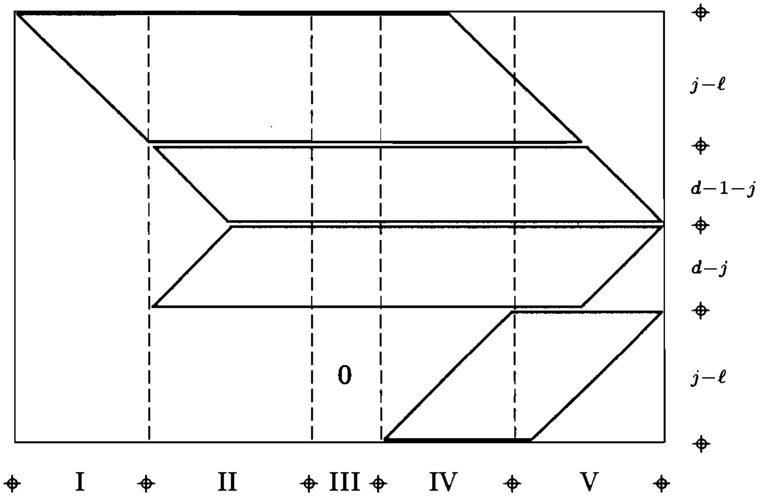
Lemma 7.10. *Assume* $\deg H_j = j > \deg H_{j-1} = k$ *and* $d_i = j > d_{i+1} = \deg G_{i+1}$. *Then the following hold.*

1. $H_{j-1} = a_d \cdot \overline{h}_j \cdot G_{i+1}$ *(so* $d_{i+1} = k$ *and* $\mathrm{lc} G_{i+1} = \dfrac{c_{j-1}}{a_d \cdot \overline{h}_j}$*),*
2. $H_\ell = 0$ *for* $k < \ell < j-1$,
3. $H_k = (-1)^{(j-k)(j-k-1)/2} \cdot a_d^{j-k} \cdot \overline{h}_j \cdot (\mathrm{lc} G_{i+1})^{j-k-1} \cdot G_{i+1}$,
4. $\overline{h}_\ell$ *is a maximal minor of* syl_ℓ *and* $h_k = \overline{h}_k$.

Proof. Assume first that $j < d$ and $\overline{h}_j = h_j$. We consider the ℓ-th Sylvester matrix syl_ℓ for $j \ge \ell \ge k$ and replace the rows of

$$BX^{d-j}, BX^{d-j+1}, \ldots, BX^{d-2-\ell}, BX^{d-1-\ell}$$

by the rows of $G_{i+1}, G_{i+1}X, \ldots, G_{i+1}X^{j-2-\ell}, G_{i+1}X^{j-1-\ell}$. By Lemma 7.9 this new matrix results from syl_ℓ through row manipulations with determinant one (adding successively linear combinations of previous rows to the rows of $BX^{d-j}, BX^{d-j+1}, \ldots, BX^{d-2-\ell}, BX^{d-1-\ell}$ of syl_ℓ) and has the following shape:



where the top parallelogram comprises the rows of

$$AX^{d-2-\ell}, AX^{d-3-\ell}, \ldots, AX^{d-j}, AX^{d-1-j},$$

the two middle parallelograms together correspond to the j-th Sylvester matrix syl_j and comprise the rows of

$$AX^{d-2-j}, AX^{d-3-j}, \ldots, AX, A$$

and of

$$B, BX, \ldots, BX^{d-2-j}, BX^{d-1-j},$$

and finally the bottom parallelogram comprises the rows of

$$G_{i+1}, G_{i+1}X, \ldots, G_{i+1}X^{j-2-\ell}, G_{i+1}X^{j-1-\ell};$$

furthermore, for the vertical stripes,

I is the block of the columns of $X^{2d-2-\ell}, X^{2d-3-\ell}, \ldots, X^{2d-j}, X^{2d-1-j}$,
II is the block of the columns of $X^{2d-2-j}, X^{2d-3-j}, \ldots, X^{j+2}, X^{j+1}$,
III is the block of the columns of $X^j, X^{j-1}, \ldots, X^{j-(\ell-k)+1}, X^{j-(\ell-k)}$,
IV is the block of the columns of $X^{j-(\ell-k)-1}, X^{j-(\ell-k)-2}, \ldots, X^{k+1}, X^k$,
V is the block of the columns of $X^{k-1}, X^{k-2}, \ldots, X, 1$.

Note that the upper triangular $(j-\ell) \times (j-\ell)$ north block of stripe I has on its diagonal the element a_d, the middle $(2d-1-2j) \times (2d-2-2j)$ block of stripe II is the matrix Σ_j (see Remark 7.4), stripe III has a $(j-\ell)\times(\ell-k+1)$ null south block, and that the $(j-\ell) \times (j-\ell)$ south block of IV has on its anti-diagonal always the element $\mathrm{lc}(G_{i+1})$.

1. For $j-\ell = 1$, the linear form $\sigma_\ell = \sigma_{j-1}$ is nonzero since $a_d \cdot h_j \neq 0$, and we find from the shape of the above matrix that $H_{j-1} = a_d \cdot h_j \cdot G_{i+1}$, which can only be zero if $G_{i+1} = 0$, that is, if $i = w$, or in other words, if $k = d_{i+1} = -1$.
   If $k = -1$ all linear forms σ_ℓ are null for $\ell < j$ since the bottom parallelogram of the above matrix is null, so we assume $k \geq 0$ in what follows.
2. For $j-\ell \geq 2$ and $\ell-k+1 \geq 2$, the null south block of III shows that the columns of I, II, and the first two of III (that is, the columns of X^j, X^{j+1}) are linearly dependent. So the same columns of Σ_ℓ are dependent, and the linear form σ_ℓ is zero. Thus $H_\ell = 0$ for $k < \ell < j-1$.
3. For $\ell = k$, stripe III consists of just the X^j-column and the shape of the above matrix shows that
$$H_k = (-1)^{(j-k)(j-k-1)/2} \cdot a_d^{j-k} \cdot h_j \cdot \mathrm{lc}(G_{i+1})^{j-k-1} \cdot G_{i+1}.$$
4. By the first statement $c_{j-1}/h_j = a_d \cdot \mathrm{lc}(G_{i+1})$, so $\overline{h}_\ell$ is the minor of the columns of stripe I, stripe II, the first column of stripe III (the X^j-column), and the columns of stripe IV of the above matrix, and hence it coincides with the same minor of syl_ℓ.

The case $j = d$ is similar and left to the reader (consider syl_ℓ directly).

Proof (of the Structure Theorem 7.5). It remains to show the third item in the Structure Theorem 7.5. Using Lemma 7.10 and relation (7.1) for the Gaussian remainder we find

$$\begin{aligned} H_{k-1} &= a_d \cdot h_k \cdot G_{i+2} = a_d \cdot h_k \cdot \mathrm{Gau}(G_i, G_{i+1}) \\ &= \frac{h_k}{\overline{h}_j} \cdot \mathrm{Gau}(G_i, a_d \cdot \overline{h}_j \cdot G_{i+1}) \\ &= \frac{h_k}{\overline{h}_j} \cdot \mathrm{Gau}(H_j, H_{j-1}) \\ &= -\frac{h_k \cdot c_{j-1}}{h_j \overline{h}_j} \cdot \mathrm{Rem}(H_j, H_{j-1}). \end{aligned}$$

Thus,

$$-h_j \overline{h}_j H_{k-1} = \mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1});$$

- if $j = d$, then

$$H_{k-1} = -\mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1}),$$

- if $j < d$, then

$$h_j^2 \cdot H_{k-1} = -\mathrm{Rem}(h_k \cdot c_{j-1} \cdot H_j, H_{j-1}).$$

The fact that this is an exact signed Euclidean division in $D[X]$ will be shown at the end of this section (Corollary 7.18).

Corollary 7.11. *For a d-couple (A, B) in $D[X]$ and $j \neq \deg(\gcd(A, B)) - 1$, we have $j \leq d - 1$,*

$$H_j = 0 \iff \sigma_j = 0 \iff \mathrm{rank}(\Sigma_j) < 2d - 2 - 2j.$$

For $j = \deg(\gcd(A, B)) - 1$, we have $H_j = 0$ and $\sigma_j \neq 0$.

The Bézout relations for the nonzero H_j of Remark 7.4 are uniquely determined through a certain degree condition on the cofactors U_j and V_j; we shall call them *the* polynomials of *the* j-th Bézout relation, that is, with definite articles.

Proposition 7.12. *For a d-couple (A, B) of polynomials in $D[X]$ the polynomials U_j and V_j in $D[X]$ in the j-th Bézout relation*

$$H_j = U_j \cdot A + V_j \cdot B$$

are uniquely determined by the degree restriction $\deg V_j \leq d - 1 - j$ for all j such that $H_j(A, B) \neq 0$. One has $\deg V_j = d - 1 - j$ if and only if $j = d$ or H_{j+1} is non-defective.

Proof. This is clear for $j = d$. For $j < d$, $\deg V_j \leq d - 1 - j$ implies the validity of the additional degree bound $\deg U_j \leq d - 2 - j$. $H_j \neq 0$ always implies $\sigma_j \neq 0$. Any such degree restricted Bézout relation for any nonzero polynomial of degree $\leq j$ uniquely corresponds to a linear form that is K-proportional to σ_j (see Remark 7.4). If this polynomial coincides with H_j, then the factor of proportionality is one. (In any case, if $\sigma_j \neq 0$, the pair (U_j, V_j) is uniquely determined up to K-proportionality by the conditions $\deg V_j \leq d - 1 - j$ and $\deg(U_j \cdot A + V_j \cdot B) \leq j$.)

Corresponding conditions make transition matrices unique; analogously to the above, we shall speak of *the* transition matrices.

Corollary 7.13. *Let* $0 \le j \le d$, *let* (A, B) *be a* d*-couple and* (C, D) *a* j*-couple of polynomials in* $K[X]$. *Assume* (C, D) *to be elementwise* K*-proportional to a couple of two consecutive signed Euclidean remainders of* (A, B), *and suppose*

$$M = \begin{pmatrix} I & J \\ K & L \end{pmatrix} \in K[X]^{2\times 2}$$

is a transition matrix, that is,

$$\begin{pmatrix} C \\ D \end{pmatrix} = M \cdot \begin{pmatrix} A \\ B \end{pmatrix}.$$

Then the conditions $\deg J \le d - 1 - j$, $D \ne 0$, *and* $\deg L \le d - j$ *imply that the transition matrix* $M = M(C, D; A, B)$ *is unique, unimodular, and that* $\deg L = d - j$.

If $D = 0$, *an additional nonzero scaling of* $\det M$ *makes* M *unique with these properties.*

Proof. By the Structure Theorem 7.5 we may assume by a monomial scaling that $(C, D) = (H_j, H_{j-1})$. Then if $H_{j-1} \ne 0$,

$$M = \begin{pmatrix} U_j & V_j \\ U_{j-1} & V_{j-1} \end{pmatrix}, \tag{7.2}$$

by Proposition 7.12. Another monomial scaling and the comparison with the unimodular signed Euclidean transition matrix (cf. Definition 2.6) show that M is unimodular with $\deg V_{j-1} = d - j$. If $H_{j-1} = 0$, then the second row of M in (7.2) may be scaled arbitrarily. Passing to the generic situation first shows that in any case $\det M \in K$ (a constant polynomial), and $H_j \ne 0$ shows $\det M \ne 0$. (For $j = 0$ pass first to the generic situation of $(d + 1)$-couples and consider $j = 1$.)

We fix a d-couple (A, B) of polynomials in $D[X]$ and study the transition between two regular couples of Sylvester-Habicht polynomials (H_j, H_{j-1}) and (H_k, H_{k-1}), that is, both regular and $0 \le k \le j \le d$. If, moreover, $H_{j-1} \sim_K H_k$, then the pair of regular couples of Sylvester-Habicht polynomials is said to be *consecutive*. For such a pair (k, j), $0 \le k \le j \le d$, we define the *Sylvester-Habicht transition matrix* by

$$N_{k,j} = N_{k,j}(A, B) = M(H_k, H_{k-1}; H_j, H_{j-1}),$$

where M is defined by Corollary 7.13. We scale $\det N_{k,j} = h_k^2/h_j^2$ if $H_{k-1} = 0$ (this is motivated by the non-final transition; see the subsequent Lemma 7.14, Proposition 7.15, and Proposition 7.16). By definition, we have

$$\begin{pmatrix} H_k \\ H_{k-1} \end{pmatrix} = N_{k,j} \cdot \begin{pmatrix} H_j \\ H_{j-1} \end{pmatrix}.$$

Now write

$$N_{k,j} = \begin{pmatrix} U_{1,k,j} & V_{1,k,j} \\ U_{2,k,j} & V_{2,k,j} \end{pmatrix}$$

with entries in $K[X]^{2\times 2}$. Note that $U_{1,k,j} = 0$ in the consecutive case.

Generally, $N_{k,j}$ is the product of all the intermediate consecutive Sylvester transition matrices.

We have the following lemma.

Lemma 7.14. *Let (A, B) be a regular d-couple of polynomials in $D[X]$, (H_j, H_{j-1}) and (H_k, H_{k-1}) two consecutive regular couples of its Sylvester-Habicht polynomials and $0 \leq k < j \leq d$. Then the anti-diagonal entries of the consecutive Sylvester-Habicht transition matrix*

$$N_{k,j} = \begin{pmatrix} 0 & V_{1,k,j} \\ U_{2,k,j} & V_{2,k,j} \end{pmatrix}$$

are:

$$V_{1,k,j} = \frac{h_k}{c_{j-1}}, \quad U_{2,k,j} = -\frac{h_k \cdot c_{j-1}}{h_j \overline{h}_j}.$$

So,

$$\det N_{k,j} = \frac{h_k^2}{h_j \overline{h}_j}.$$

Moreover, $\deg V_{2,k,j} = j - k$.

Proof. Immediate consequences of points 2 and 3a) of the reformulation of Structure Theorem 7.5 in Remark 7.6.

Let $d = d_0 > d_1 > \ldots > j = d_i > \ldots > d_w$ denote the degree sequence in the remainder sequence of the d-couple (A, B). We consider the Sylvester-Habicht transition matrix $N_{j,d} = N_{j,d}(A, B)$,

$$N_{j,d} = N_{d_i,d_{i-1}} \cdots N_{d_1,d_0}$$

of an 'absolute transition' satisfying

$$\begin{pmatrix} H_j \\ H_{j-1} \end{pmatrix} = N_{j,d} \cdot \begin{pmatrix} H_d \\ H_{d-1} \end{pmatrix} = N_{j,d} \cdot \begin{pmatrix} A \\ B \end{pmatrix}.$$

Proposition 7.15. *Let (A, B) be a d-couple of polynomials in $D[X]$, $0 \leq j \leq d$, and (H_j, H_{j-1}) a regular couple of its Sylvester-Habicht polynomials. Then*

$$N_{j,d} = \begin{pmatrix} U_j & V_j \\ U_{j-1} & V_{j-1} \end{pmatrix},$$

where $U_j, V_j, U_{j-1}, V_{j-1} \in D[X]$ are the polynomials of the j-th and $(j-1)$-th Bézout relations.

Proof. By Proposition 7.12 and Corollary 7.13.

For a 'relative transition' one has the following result.

Proposition 7.16. *Let (A, B) be a d-couple of polynomials in $D[X]$, (H_j, H_{j-1}) and (H_k, H_{k-1}) two regular couples of its Sylvester-Habicht polynomials and $0 \le k \le j \le d$. Then*

$$\det N_{k,j} = \frac{h_k^2}{h_j \overline{h}_j},$$

$$h_j \overline{h}_j \cdot N_{k,j} = \begin{pmatrix} U_k & V_k \\ U_{k-1} & V_{k-1} \end{pmatrix} \cdot \begin{pmatrix} V_{j-1} & -V_j \\ -U_{j-1} & U_j \end{pmatrix},$$

and therefore the elements of this scaled matrix lie in $D[X]$.

Proof. The matrix $N_{k,j}$ is the product of all intermediate consecutive Sylvester-Habicht transition matrices; so, by the multiplicativity property of determinants and Lemma 7.14, $\det N_{k,j} = h_k^2 / h_j \overline{h}_j$.

When $j \neq d$, the factorizations of $N_{k,j}$ and of $N_{j,d}$ show that we can in fact write $N_{k,j} = N_{k,d} \cdot N_{j,d}^{-1}$; so, by Proposition 7.16 and $\det N_{j,d} = h_j \overline{h}_j$, the scaled matrix is in fact free of denominators.

Corollary 7.17. $V_{2,k,j} = \frac{-U_{k-1} \cdot V_j + V_{k-1} \cdot U_j}{h_j \overline{h}_j}$.

Next, we deduce the exact divisibility stated in Structure Theorem 7.5.

Corollary 7.18.

$$-h_k \cdot c_{j-1} \cdot H_j = (U_{k-1} \cdot V_j - V_{k-1} \cdot U_j) \cdot H_{j-1} + h_j \overline{h}_j \cdot H_{k-1}$$

is an exact signed Euclidean division for $j \le d$.

Proof. The exact divisibility for $j = d$ is obvious. For $j < d$, it is a consequence of the preceding result.

We end this section with the proof of Proposition 2.16.

Proof. (of Proposition 2.16) The proposition is clear from the definition of the polynomials in the Sylvester-Habicht sequence in terms of determinants.

A more detailed study of the specialization properties of the Sylvester-Habicht sequence can be found in [19].

7.3 Sylvester-Habicht Sequences and Cauchy Index

We first prove Theorem 2.17, using the preceding Structure Theorem.

Proof. (of Theorem 2.17) We can suppose without loss of generality that (A, B) is a regular d-couple. If $x_1 < \ldots < x_r$ are the real roots of the non identically zero polynomials

$$H_j = \mathrm{SyHa}_j(A, B) \quad j \in \{0, \ldots, d\},$$

with $h_j = \mathrm{syha}_j(A, B)$, we write $x_0 < x_1 < \ldots < x_r < x_{r+1}$ with x_0 (respectively x_{r+1}) sufficiently small (respectively big). For every $i \in \{1, \ldots, r\}$ choose an element y_i between x_i and x_{i+1}. Take $y_0 = x_0$ and $y_r = x_{r+1}$.

We have the following equalities:

$$\begin{aligned} V_{\mathrm{SyHa}}(A, B) &= V_{\mathrm{SyHa}}(A, B; -\infty) - V_{\mathrm{SyHa}}(A, B; +\infty) \\ &= \sum_{i=1}^{r} \Big[V_{\mathrm{SyHa}}(A, B; y_{i-1}) - V_{\mathrm{SyHa}}(A, B; y_i) \Big], \end{aligned}$$

which reduces the proof of the theorem to the study of the integer:

$$V_{\mathrm{SyHa}}(A, B; y_{i-1}) - V_{\mathrm{SyHa}}(A, B; y_i)$$

for every $i \in \{1, \ldots, r\}$.

We remark that those polynomials in the list $[H_0, \ldots, H_d]$, which are not identically 0 do not vanish at any y_i and that H_ℓ, the last polynomial which is not identically 0 in the Sylvester-Habicht sequence, is a greatest common divisor of A and B. Consequently, if x is not a root of A, $H_\ell(x) \neq 0$. The proof of the theorem is based on the following lemmas.

Lemma 7.19. *If* $A(x_i) \neq 0$ *and* $H_{j-1}(x_i) = 0$ *with* $\deg(H_{j-1}) = j - 1$, *then, for* $y \in \{y_{i-1}, y_i\}$*:*

$$\mathrm{sign}(H_{j-2}(y) H_j(y)) = -1,$$

so that, if $j \leq d - 1$, *then*

$$V([H_j, H_{j-1}, H_{j-2}]; y_{i-1}) = V([H_j, H_{j-1}, H_{j-2}]; y_i) = 1.$$

Proof. Since $H_{j-1}(x_i) = 0$ and $H_\ell(x_i) \neq 0$, we have $H_{j-2}(x_i) \neq 0$. So applying identity 3 in Theorem 7.5 and using Remark 7.6 we get

$$h_j \overline{h}_j H_{j-2} = -h_{j-1}^2 \mathrm{Rem}(H_{j+1}, H_j) \Longrightarrow \mathrm{sign}(H_{j-2}(x_i) H_j(x_i)) = -1,$$

which implies, if $j \leq d - 1$, regardless of the sign of $H_{j-1}(y)$ with $y \in \{y_{i-1}, y_i\}$,

$$V([H_j, H_{j-1}, H_{j-2}]; y_{i-1}) = V([H_j, H_{j-1}, H_{j-2}]; y_i) = 1.$$

Lemma 7.20. *If $A(x_i) \neq 0$ and $H_{j-1}(x_i) = 0$, with $\deg(H_{j-1}) = k < j-1$, then, for $y \in \{y_{i-1}, y_i\}$:*

$$\operatorname{sign}(H_{k-1}(y)H_j(y)) = -\operatorname{sign}(H_k(y)H_{j-1}(y))$$

so that, if $j \leq d-1$, then, for $y \in \{y_{i-1}, y_i\}$:

$$V([H_j, H_{j-1}, H_k, H_{k-1}]; y) = \begin{cases} 2 & \textit{if } H_j(x_i)H_{k-1}(x_i) > 0 \\ 1 & \textit{if } H_j(x_i)H_{k-1}(x_i) < 0. \end{cases}$$

Proof. Again since $H_\ell(x_i) \neq 0$, $H_j(x_i) \neq 0$. We denote by c_{j-1} the leading coefficient of H_{j-1}. Applying Identity 3 in Theorem 7.5, and Remark 7.6 we get: $h_j\overline{h}_j H_{k-1}(x_i) = -c_{j-1}h_k H_j(x_i)$ so that $H_{k-1}(x_i) \neq 0$, and using identity 1 in Theorem 7.5 we see that, for $y \in \{y_{i-1}, y_i\}$

$$\operatorname{sign}(H_{k-1}(y)H_j(y)) = -\operatorname{sign}(H_k(y)H_{j-1}(y)),$$

from which, looking at all possible cases, we derive:

$$V([H_j, H_{j-1}, H_k, H_{k-1}]; y) = \begin{cases} 2 & \text{if } H_{k-1}(x_i)H_j(x_i) > 0 \\ 1 & \text{if } H_{k-1}(x_i)H_j(x_i) < 0, \end{cases}$$

so that in all cases

$$V([H_j, H_{j-1}, H_k, H_{k-1}]; y_{i-1}) = V([H_j, H_{j-1}, H_k, H_{k-1}]; y_i).$$

Lemma 7.21. *If $i \in \{1, \ldots, r\}$ and $A(x_i) \neq 0$, then*

$$V_{\mathrm{SyHa}}(A, B; y_{i-1}) - V_{\mathrm{SyHa}}(A, B; y_i) = 0.$$

Proof. We note first that since $A(x_i) \neq 0$, the signs of H_d, H_ℓ and H_k –with $H_k(x_i) \neq 0$– in y_{i-1} and y_i coincide. So, we need only know the behaviour of the polynomials that are not identically 0 in the sequence when x_i is a root of some H_j. The only two possibilities are the following.

1. $j \leq d-1$ with $\deg(H_{j-1}) = j-1$, $\deg(H_j) = j$ and $H_{j-1}(x_i) = 0$: according to Lemma 7.19,

$$V([H_j, H_{j-1}, H_{j-2}]; y_{i-1}) = V([H_j, H_{j-1}, H_{j-2}]; y_i).$$

2. $j \leq d-1$ with $k = \deg(H_{j-1}) < j-1$, $\deg(H_j) = j$ and $H_{j-1}(x_i) = 0$: in this case, according to Lemma 7.20,

$$V([H_j, H_{j-1}, H_k, H_{k-1}]; y_i) = V([H_j, H_{j-1}, H_k, H_{k-1}]; y_{i-1}).$$

Thus, we conclude that if x_i satisfies $A(x_i) \neq 0$, it follows that

$$V_{\mathrm{SyHa}}(A, B; y_{i-1}) - V_{\mathrm{SyHa}}(A, B; y_i) = 0,$$

which is what we wanted to show.

Lemma 7.22. *If* $i \in \{1, \dots, r\}$ *and* $A(x_i) = 0$, *then*

$$V_{\text{SyHa}}(A, B; y_{i-1}) - V_{\text{SyHa}}(A, B; y_i) = \varepsilon_{x_i}$$

(cf. Definition 2.1).

Proof. Let $d = \deg(A) > q = \deg(B)$. Let ℓ be such that $H_\ell \neq 0$, $H_j = 0$ for every $j < \ell$. We define a new sequence $\mathcal{K} = [K_{d-\ell}, \dots, K_0]$ of polynomials: let H be the monic polynomial proportional to H_ℓ and define

$$K_j = H_{j+\ell}/H \qquad j \in \{0, \dots, d - \ell\}.$$

The first observation is that the Cauchy index of $K_{d-\ell}/K_{d-\ell-1}$ coincides with the Cauchy index of B/A. Clearly we also have

$$V([K_{d-\ell}, \dots, K_0]; y) = V([H_d, \dots, H_\ell]; y)$$

with $y \in \{y_i, y_{i-1}\}$.

For every j in $\{0, \dots, d - \ell - 1\}$ we define

$$d_j = h_{j+\ell}.$$

Then for every $j \in \{0, \dots, d-\ell\}$ such that $d_j \neq 0$ and $\deg(K_{j-1}) = k \leq j-1$, the sequence $\mathcal{K}$ has the following properties, denoting by c'_{j-1} the leading coefficient of K_{j-1}:

1. $c'_{j-1} K_k = d_k K_{j-1}$,
2. if $k < \ell < j - 2$, then $K_\ell = 0$,
3. $d_j^2 K_{k-1} = -c'_{j-1} d_k \text{Rem}(K_j, K_{j-1})$,

as an easy consequence of Theorem 7.5, given the definition of $\mathcal{K}$.

We can write:

$$A(X) = (X - x_i)^e p(X) \qquad p(x_i) \neq 0,$$

$$B(X) = (X - x_i)^f q(X) \qquad q(x_i) \neq 0,$$

and we only need to study two cases:

1. $f \geq e$. In this case $K_{d-\ell}(x_i) \neq 0$ and we can proceed as in Lemma 7.21 using the properties of the polynomials in the sequence $\mathcal{K}$ which are the same as the polynomials in the Sylvester-Habicht sequence.
2. $f < e$. In this case $K_{d-\ell}(x_i) = 0$ and $K_{d-\ell-1}(x_i) \neq 0$. Proceeding as in Lemma 7.21 and using the properties of $\mathcal{K}$, we conclude that $V([K_{d-\ell-1}, \dots, K_0]; y_{i-1}) - V([K_{d-\ell)-1}, \dots, K_0]; y_i) = 0$. So we only need to study $V([K_{d-\ell}, K_{d-\ell-1}]; y_{i-1}) - V([K_{d-\ell}, K_{d-\ell-1}]; y_i)$ when $K_{d-\ell-1}(x_i) \neq 0$:

$$K_{d-\ell}(X) = (X - x_i)^{e-f} \underline{p}(X) \qquad \underline{p}(x_i) \neq 0,$$

$$K_{d-\ell-1}(X) = \underline{q}(X) \qquad \underline{q}(x_i) \neq 0,$$

which gives, with a look at all possible cases,

$$V([K_{d-\ell},K_{d-\ell-1}];y_{i-1}) - V([K_{d-\ell},K_{d-\ell-1}];y_i) =$$

$$= \left\{ \begin{array}{ll} 0 & \text{if } e-f \text{ is even} \\ 1 & \text{if } e-f \text{ is odd and } p(x_i)q(x_i)>0 \\ -1 & \text{if } e-f \text{ is odd and } p(x_i)q(x_i)<0 \end{array} \right\} = \varepsilon_{x_i}.$$

This ends the proof of Theorem 2.17.

We now prove Proposition 2.20, using again the Structure Theorem.

Proof (Proof of Proposition 2.20). We denote $H_j = \mathrm{SyHa}_j(A,B)$ and $h_j = \mathrm{syha_j}(A,B)$. If all of the polynomials in the Sylvester-Habicht sequence associated to A and B are *regular* (i.e., their index in the Sylvester-Habicht sequence is equal to their degree), then it is clear that:

$$\begin{aligned} I(B/A) &= V_{\mathrm{SyHa}}(A,B) \\ &= V([(-1)^d h_d,(-1)^{d-1}h_{d-1},\dots,-h_1,h_0]) - V([h_d,\dots,h_1,h_0]) \\ &= D([h_0,\dots,h_d]). \end{aligned}$$

where V denotes the number of sign variations in the considered list.

Problems arise when there are defective polynomials in the Sylvester-Habicht sequence, i.e., when there appears some h_{j-1} that is equal to zero. In this case, the principal Sylvester-Habicht coefficient is not the leading coefficient of H_{j-1}. Denote as usual by k the degree of H_{j-1}. The situation is as follows:

$$h_j \neq 0, h_{j-1}=0,\dots,h_{k+1}=0, h_k \neq 0.$$

We need to prove the following equality:

$$\tau \overset{\mathrm{def}}{=} V([H_j,H_{j-1},H_k];-\infty) - V([H_j,H_{j-1},H_k];+\infty)$$

$$= \sigma \overset{\mathrm{def}}{=} \left\{ \begin{array}{ll} 0 & \text{if } j-k \text{ is even} \\ (-1)^{(j-k-1)/2}\cdot \mathrm{sign}(h_k h_j) & \text{if } j-k \text{ is odd.} \end{array} \right.$$

Applying Theorem 7.5 to this situation, we first notice that H_{j-1} and H_k are proportional, so that $\mathrm{sign}(H_{j-1}H_k,+\infty) = \mathrm{sign}(H_{j-1}H_k,-\infty)$. We denote by σ_j, σ_{j-1} and σ_k, respectively, $\mathrm{sign}(H_j,+\infty)$, $\mathrm{sign}(H_{j-1},+\infty)$ and $\mathrm{sign}(H_k,+\infty)$.

- When $j-k$ is even, the sequence of signs of $[H_j,H_{j-1},H_k]$ at $+\infty$ is $[\sigma_j,\sigma_{j-1},\sigma_k]$ and at $-\infty$ is $[\sigma_j,\sigma_{j-1},\sigma_k]$ when j is even (respectively $[-\sigma_j,-\sigma_{j-1},-\sigma_k]$ when j is odd), which implies that $\tau=0$ when $j-k$ is even.

- When $j-k$ is odd, the sequence of signs of $[H_j, H_{j-1}, H_k]$ at $+\infty$ is $[\sigma_j, \sigma_{j-1}, \sigma_k]$ and at $-\infty$ is $[\sigma_j, -\sigma_{j-1}, -\sigma_k]$ when j is even (respectively $[-\sigma_j, \sigma_{j-1}, \sigma_k]$ when j is odd). Also, using Theorem 7.5, 1, we see that

$$\sigma_k = (-1)^{\frac{(j-k)(j-k-1)}{2}} \sigma_{j-1}$$

so that $\sigma = \text{sign}(h_k h_j)$, when $(j-k-1)/2$ is even, because

$$(-1)^{\frac{(j-k)(j-k-1)}{2}} = 1,$$

and $\sigma = -\text{sign}(h_k h_j)$ when $(j-k-1)/2$ is odd, because

$$(-1)^{\frac{(j-k)(j-k-1)}{2}} = -1.$$

The equality between σ and τ is now obtained in all cases:

- if $j-k$ is even, then $\tau = \sigma = 0$,
- if $j-k$ is odd and $(j-k-1)/2$ is even, then $\tau = \sigma = \text{sign}(h_k h_j)$,
- if $j-k$ is odd and $(j-k-1)/2$ is odd, then $\tau = \sigma = -\text{sign}(h_k h_j)$.

References

1. Basu, S., Pollack, R., and Roy, M.-F. (1996): *On the combinatorial and algebraic complexity of Quantifier elimination.* J. Assoc. Comput. Machin. **43**, 1002–1045.
2. Becker, E.: *Sums of Squares and Trace Forms in Real Algebraic geometry.* Cahiers du Séminaire d'Histoire des Mathématiques, 2 ème série, Vol. 1 (1991), Université Pierre et Marie Curie.
3. Becker, E. and Woermann, T.: *On the trace formula for quadratic forms.* Jacob, William B. (ed.) et al., Recent advances in real algebraic geometry and quadratic forms. Proceedings of the RAGSQUAD year, Berkeley. Contemp. Math. **155**, 271–291 (1994).
4. Ben-Or, M., Kozen, D., and Reif, J. (1986): *The complexity of elementary algebra and geometry.* J. Comput. Syst. Sci. **32**, 251–264.
5. Bochnak J., Coste, M., and Roy, M.-F. *Géométrie algébrique réelle* (1987): (*Real algebraic geometry*). Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge, Bd. 12, Springer-Verlag, Berlin Heidelberg New York.
6. Brown, W.S. (1971): *On Euclid's algorithm and the computation of polynomial greatest common divisors.* J. Assoc. Comput. Machin. **18**, 478–504.
7. Brown, W.S. and Traub, J.F. (1971): *On Euclid's algorithm and the theory of subresultants.* J. Assoc. Comput. Machin. **18**, 505–514
8. Collins, G.E. (1967): *Subresultants and reduced polynomial remainder sequences.* J. Assoc. Comput. Mach. **14**, 128–142.
9. Collins, G.E. (1975): *Quantifier elimination for real closed fields by cylindrical algebraic decomposition.* Autom. Theor. form. Lang., 2nd GI Conf., Kaiserslautern 1975, Lect. Notes Comput. Sci. **33**, 134–183.
10. Collins, G.E. and Loos, R. (1982): *Real zeroes of polynomials.* Computer Algebra, Symbolic and Algebraic Computation, Comput. Suppl. **4**, 83–94.

11. Coste, M. and Roy, M.-F. (1988): *Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets.* J. Symb. Comput. **5**, No. 1/2, 121–129.
12. Cucker, F., Gonzalez-Vega, L., and Rossello, F. (1991): *On algorithms for real algebraic plane curves.* Effective Methods in Algebraic Geometry, Proc. Symp., Castiglioncello/Italy 1990, Prog. Math. **94**, 63–87.
13. Davenport, J. and Heintz, J. (1988): *Real Quantifier Elimination is Doubly Exponential.* J. Symb. Comput. **5**, No. 1/2, 29–36.
14. Demazure, M. (1985): *Charles Hermite: déjà ....* Notes informelles de calcul formel VII. Ecole Polytechnique.
15. Ducos, L. (1996): *Algorithme de Bareiss, Algorithme des sous-résultants.* RAIRO, Inf. Theor. Appl. **30**, No. 4, 319–347.
16. Gantmacher, F. R. (1966): *Théorie des matrices.* Vol. 2, Dunod.
17. Gonzalez Vega, L. (1989): *La sucesión de Sturm–Habicht y sus aplicaciones al Algebra Computacional.* Doctoral Thesis. Universidad de Cantabria.
18. Gonzalez-Vega, L., Lombardi, H., Recio, T., and Roy, M.-F. (1989): *Sturm-Habicht sequence.* Proceedings of ISSAC-89 (Portland), 136–146, ACM-Press.
19. Gonzalez-Vega, L., Lombardi, H., Recio, T., and Roy, M.-F. (1994): *Specialisation de la suite de Sturm et sous-resultants (Specialization of the Sturm sequence and subresultants).* I) RAIRO, Inf. Theor. Appl. **24**, No. 6, 561–588 (1990). II.) RAIRO, Inf. Theor. Appl. **28**, 1–24.
20. Gonzalez-Vega, L., Lombardi, H., Recio, T., and Roy, M.-F. (1998): *Determinants and real roots of univariate polynomials.* Quantifier Elimination and Cylindrical Algebraic Decomposition (B. F. Caviness and J. R. Johnson, eds.), Texts and Monographs in Symbolic Computation, 300–316, Springer-Verlag, Wien New York.
21. Gonzalez-Vega, L. (1996): *A combinatorial algorithm solving some quantifier elimination problems.* Quantifier Elimination and Cylindrical Algebraic Decomposition (B. F. Caviness and J. R. Johnson, eds.), Texts and Monographs in Symbolic Computation, 365–375, Springer-Verlag, Wien New York.
22. Gonzalez-Vega, L. (1996): *A special quantifier elimination algorithm for Pham systems.* Preprint, Santander.
23. Grigor'ev, D. Yu. (1988): *Complexity of deciding Tarski algebra.* J. Symb. Comput. **5**, No. 1/2, 65–108.
24. Grigor'ev, D. Yu. and Vorobjov, N. N. (1988): *Solving systems of polynomial inequalities in subexponential time.* J. Symb. Comput. **5**, No. 1/2, 37–64.
25. Habicht, W. (1948): *Eine Verallgemeinerung des Sturmschen Wurzelzaehlverfahrens.* Comment. Math. Helv. **21**, 99–116.
26. Heintz, J., Roy, M.-F., and Solerno, P. (1990): *Sur la complexité du Principe de Tarski-Seidenberg (On the complexity of the Tarski-Seidenberg principle).* Bull. Soc. Math. Fr. **118**, No. 1, 101–126.
27. Heintz, J., Roy, M.-F., and Solerno, P. (1993): *On the theoretical and practical complexity of the existential theory of the reals.* Comput. J. **36**, No. 5, 427–431.
28. Hong, H. (1993): *Quantifier elimination for formulas constrained by quadratic equations via slope resultants.* Computer Journal **36**, 5, 439–449.
29. D. Lazard (1997): *Sous-résultants,* unpublished manuscript, Paris.
30. Lickteig, T. and Roy, M.-F. (1996): *Cauchy index computation.* Calcolo **33**, 337–351.
31. Lickteig, T. and Roy, M.-F. (1997): *Sylvester-Habicht sequences and fast Cauchy index computation.* Preprint, Rennes.
32. Lombardi, H. (1989): *Algèbre Elémentaire en temps polynomial.* Doctoral Thesis, University of Nice.
33. Loos, R. (1982): *Generalized polynomial remainder sequences.* Computer Algebra, Symbolic and Algebraic Computation, Comput. Suppl. **4**, 115–137.

34. Mignotte, M. (1982): *Some useful bounds*. Computer Algebra, Symbolic and Algebraic Computation, Comput. Suppl. **4**, 259–263.
35. Noonburg, V. W. (1989): *A neural network modeled by an adaptive Lotka-Volterra system*. SIAM Journal on Applied Mathematics **49**, 1779–1792.
36. Pedersen, P. (1991): *Couting real zeroes*, Thesis, Courant Institute, New York University.
37. Pedersen, P., Roy, M.-F., and Szpirglas, A. (1993): *Counting real zeros in the multivariate case*. Computational Algebraic Geometry, Progress in Mathematics **109**, 203–224, Birkhäuser, Basel.
38. Quitté, C. (1997): *Une démonstration de l'algorithme de Bareiss par l'algèbre extérieure*. Unpublished manuscript.
39. Renegar, J. (1992): *On the computational complexity and geometry of the first-order theory of the reals*. Parts I, II and III. J. Symb. Comput. **13** (3) 255–352.
40. Renegar, J. (1991): *Recent progress on the complexity of the decision problem for the reals*. Discrete and computational geometry, Proc. DIMACS Spec. Year Workshops 1989-90, DIMACS, Ser. Discret. Math. Theor. Comput. Sci. **6**, 287–308.
41. Rouillier, F. (1995): *Formules de Bareiss et reduction de formes quadratiques. (Bareiss formulas and reduction of quadratic forms)*. C. R. Acad. Sci. Paris, Ser. I, 320, 10, 1273–1277.
42. Roy, M.-F. (1996): *Basic algorithms in real algebraic geometry: from Sturm theorem to the existential theory of reals*. Lectures on Real Geometry in memoriam of Mario Raimondo, Expositions in Mathematics **23**, 1–67. de Gruyter, Berlin.
43. Roy, M.-F. and Szpirglas, A. (1990): *Complexity of computation on real algebraic numbers*. J. Symb. Comput. **10**, No. 1, 39–51.
44. Seidenberg, A. (1954): *A new decision method for elementary algebra*. Ann. Math. **60**, 365–374.
45. Tarski, A. (1951): *A decision method for elementary algebra and geometry*. Prepared for publication by J.C.C. Mac Kinsey.
46. Weispfenning, V. (1995): *Solving parametric polynomial equations and inequalities by symbolic algorithms*. Computer Algebra in Science and Engineering, 163–179, World Scientific.
47. Weispfenning, V. (1997): *Quantifier elimination for real algebra – the quadratic case and beyond*. J. of AAECC **8**, 85–101.
48. Weispfenning, V. (1995): *A new approach to quantifier elimination for real algebra*. Quantifier Elimination and Cylindrical Algebraic Decomposition (B. F. Caviness and J. R. Johnson, eds.), Texts and Monographs in Symbolic Computation, 376–392, Springer-Verlag, Wien New York.

Chapter 7. Gröbner Bases and Integer Programming

Günter M. Ziegler

1. Introduction

'Integer programming' is a basic mathematical problem, of central importance in Optimization and Operations Research. While a systematic body of theory has been developed for it in the last fifty years [14], it has been realized only very recently, first by Conti & Traverso [5], that the Buchberger algorithm (cf. Chapter 1) provides a solution strategy for integer programming problems, in particular in the case of families of programs with 'varying right hand side'.

Section 2 gives a short introduction to 'what integer programming is about'. Then we discuss a basic version of the Buchberger algorithm applied to integer programming (Section 3). In Section 4 we show how, in the special case of the binomial ideals that arise from integer programs, the Buchberger algorithm can be formulated as a combinatorial-geometric algorithm that operates on lattice vectors. A surprisingly simple variation of the Buchberger algorithm for integer programming is presented in Section 5.

The problems to this chapter treat the relation between lattice vectors and binomials, and the 'Gröbner basis of a lattice', in more detail.

2. What is Integer Programming?

A *polyhedron* P is any intersection $P := \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ of closed halfspaces, in some $\mathbb{R}^n$. Here $A \in \mathbb{R}^{m \times n}$ is a matrix, while $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ are column vectors. A bounded polyhedron is a *polytope.*

Given any linear function $\mathbf{x} \longmapsto \mathbf{c}^\top \mathbf{x}$ on $\mathbb{R}^n$, the *linear programming problem* is to determine a point $\mathbf{x}$ in the polyhedron P for which the linear function $\mathbf{c}^\top \mathbf{x}$ is minimal. The points in P are called *feasible.* If P is a nonempty polytope, then the existence of an *optimal* point $\mathbf{x}_0$ is guaranteed. Furthermore, if the linear function $\mathbf{c}^\top$ is *generic* (with respect to the inequalities that define P), then the optimal point $\mathbf{x}_0$ is unique.

It is both useful and customary to deal only with a restricted class of rational polyhedra in some 'standard form'. That is, one considers, for example, only polyhedra of the form

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

with the additional assumptions that A and $\mathbf{b}$ have only integral coordinates. This is not much of a loss of generality: nonrational data do not occur 'in nature,' and general polyhedra can be represented by polyhedra in this *equality standard form* by suitable coordinate transformation, introduction of slack variables, multiplication by common denominators, etc.

Linear programming is a well-established theory. In a practical sense the linear programming problem is 'solved': after essential progress in the eighties (including the construction of polynomial algorithms such as the 'ellipsoid method', the rise of 'interior point methods' that are both theoretically polynomial and practically competitive, and considerable refinements of the classical 'simplex method'), we have now codes available (such as CPLEX, by Bob Bixby) which will solve virtually every linear program that you can cook up and store in a computer.

The situation is vastly different for *integer programming*, the task to compute an *integral* vector in P that minimizes $\mathbf{c}^\top\mathbf{x}$. In this situation, the points in $P \cap \mathbb{Z}^n$ are called *feasible*. A basic result of polyhedral theory states that the convex hull

$$P_I \quad = \quad \operatorname{conv}\{\mathbf{x} \in \mathbb{Z}^n \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

is also a polyhedron with finitely many facets. However, the facet-defining inequalities for the polyhedron P_I are not usually known in general (otherwise we would be done by linear programming), they are hard to determine, and their number may be huge. Thus the integer programming problem — of essential importance in many practical applications — is still a great challenge. It is still not difficult to produce integer programs of reasonable size ($m = 20$ and $n = 30$, say) that none of the currently available codes can solve. Here *solution*, as we will see, really comprises two separate tasks, both of them nontrivial: to *find* an optimal solution, and to *prove* that it is optimal.

While several 'good', or at least 'interesting', solution strategies exist, integer programming is still a difficult problem. The object of this chapter is an introduction to one (relatively new) such strategy: the construction of *test sets* for integer programming via the Buchberger algorithm. In a basic version, we will present this in the following section.

3. A Buchberger Algorithm for Integer Programming

For the following exposition, we consider a family of integer programs for which $A \in \mathbb{N}^{m\times n}$ is a fixed, nonnegative integer matrix. The right hand side vector, $\mathbf{b} \in \mathbb{N}^m$, is considered as variable. Thus we consider the integer polyhedra

$$P_I(\mathbf{b}) \quad := \quad \operatorname{conv}\{\mathbf{x} \in \mathbb{N}^n \mid A\mathbf{x} = \mathbf{b}\}.$$

Now let $\mathbf{c} \neq \mathbf{0}$ be a (fixed) linear objective function. To make life easier, we also assume that the linear program

$$LP(\mathbf{b}) \qquad \min \mathbf{c}^\top\mathbf{x} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$$

is bounded for every $\mathbf{b}$. This is not much of a restriction: for example, it is satisfied if $\mathbf{c} \geq \mathbf{0}$. In particular, it implies that the integer programs

$$\mathrm{IP}(\mathbf{b}) \qquad \min \mathbf{c}^\top\mathbf{x} : \quad A\mathbf{x} = \mathbf{b},\ \mathbf{x} \in \mathbb{N}^n$$

are bounded.

In the following, we also need that the objective function is generic. To enforce this, we choose a term order $\prec$ (lexicographic, for example) that can be used as a tie breaker, and define

$$\mathbf{x} \prec_{\mathbf{c}} \mathbf{y} \qquad :\Longleftrightarrow \qquad \begin{cases} \mathbf{c}^\top\mathbf{x} < \mathbf{c}^\top\mathbf{y}, & \text{or} \\ \mathbf{c}^\top\mathbf{x} = \mathbf{c}^\top\mathbf{y} & \text{and } \mathbf{x} \prec \mathbf{y}. \end{cases}$$

We will use a total order such as $\prec_{\mathbf{c}}$, derived from the 'original' objective function, as the input for the integer programming algorithms of this chapter. Note that $\prec_{\mathbf{c}}$ is a term order (in the sense of Gröbner basis theory) if and only if $\mathbf{c} \geq \mathbf{0}$.

With the above assumptions, we get that each of the integer programs

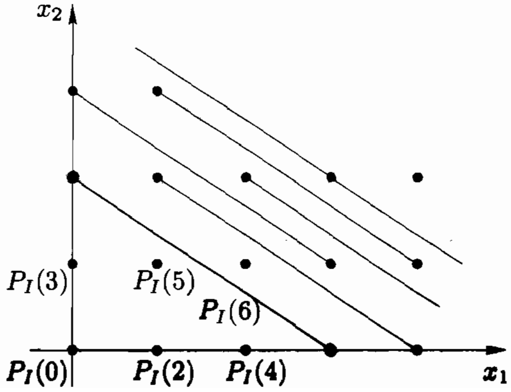
$$\mathrm{IP}(\mathbf{b}) \qquad \min_{\prec_{\mathbf{c}}} \{\mathbf{x} \in \mathbb{N}^n \mid A\mathbf{x} = \mathbf{b}\}$$

either has no feasible points, or it has a unique optimal solution. (The optimal solution, but not its objective function value, will in general depend on the tie breaker used to define $\prec_{\mathbf{c}}$.) We use

$$\mathrm{IP}_{A,\mathbf{c}}$$

to denote the whole family of these integer programs, with fixed A and $\mathbf{c}$, but varying right hand side $\mathbf{b}$. The key idea is to consider this whole class of programs simultaneously.

Example 3.1. For $n = 2$ we can draw figures such as the following, which is obtained for $A = (2\ 3)$, where $P_I(b_1)$ denotes the convex hull of the feasible (integer) points of $\mathrm{IP}(\mathbf{b})$, for $\mathbf{b} = (b_1)$. We get that $P_I(0) = \emptyset$, the set $P_I(b_1)$ is a point for $b_1 \in \{0, 2, 3, 4, 5, 7\}$, while $P_I(b_1)$ is a (bounded) line segment for $b_1 = 6$ and for $b_1 \geq 8$.



We call a vector $\mathbf{x} \in \mathbb{N}^n$ *non-optimal* if there is another vector $\mathbf{y} \in \mathbb{N}^n$ that is feasible for the same right hand side (that is, $A\mathbf{x} = A\mathbf{y}$), and that is 'better' than $\mathbf{x}$ in the sense that $\mathbf{y} \prec_{\mathbf{c}} \mathbf{x}$. A simple but crucial observation is that if $\mathbf{x}$ is non-optimal, and if $\mathbf{x}' \geq \mathbf{x}$ is componentwise larger than $\mathbf{x}$, then $\mathbf{x}'$ is non-optimal as well. Thus the Gordan-Dickson lemma, according to which every subset of $\mathbb{N}^n$ has only finitely many minimal elements (for the componentwise order), yields the following key fact.

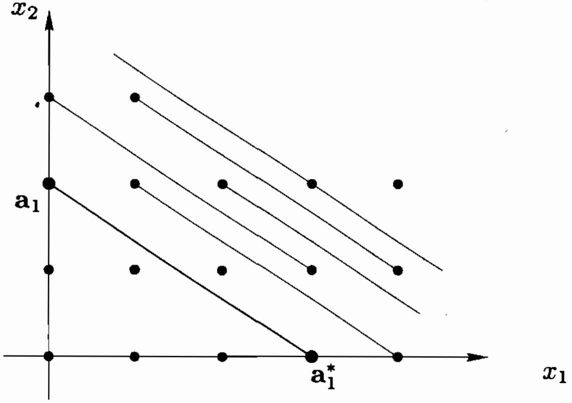
Lemma 3.2 (Minimal Non-optimal Points). *The minimal (with respect to inclusion) set of vectors $\mathbf{a}_i \in \mathbb{N}^n$ such that*

$$\{\mathbf{x} \in \mathbb{N}^n \mid \mathbf{x} \textit{ non-optimal}\} \quad = \quad \{\mathbf{x} \in \mathbb{N}^n \mid \mathbf{x} \geq \mathbf{a}_i \textit{ for some } i\}$$

is unique and finite, and thus it can be written as $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_t\}$.

This lemma is important since it yields the indexing set for both the minimal test set for $\mathrm{IP}_{A,\mathbf{c}}$, as follows, and for the Gröbner basis of the associated ideal, see Theorem 3.7 below.

Example 3.3. For $n \leq 2$ we necessarily have $t = 1$ (Exercise!). Our figure depicts the situation for $n = 2$, $A = (2\ 3)$, $\mathbf{b} = (6)$, and $\mathbf{c} = (1\ 4)$. The shaded region covers all the non-optimal integral points for this family of programs.



Here both $\mathbf{a}_1$ and $\mathbf{a}_1^*$ are contained in $P_I(\mathbf{b})$, for $\mathbf{b} = A\mathbf{a}_1 = A\mathbf{a}_1^* = (6)$.

Definition 3.4. A subset $\mathcal{G}_{\mathbf{c}} \subseteq \mathbb{Z}^n$ is a *test set* for the family $\mathrm{IP}_{A,\mathbf{c}}$ of integer programs if and only if

- $A\mathbf{g} = \mathbf{0}$ for all $\mathbf{g} \in \mathcal{G}_{\mathbf{c}}$,
- $\mathbf{g} \succ_{\mathbf{c}} \mathbf{0}$ for all $\mathbf{g} \in \mathcal{G}_{\mathbf{c}}$, and
- for every non-optimal point $\mathbf{x} \in \mathbb{N}^n$, there is some $\mathbf{g} \in \mathcal{G}_{\mathbf{c}}$ with $\mathbf{x} - \mathbf{g} \geq \mathbf{0}$.

The definition of a test set immediately provides us with the following algorithm for integer programming — once we have a feasible point to start with, and we know how to compute a test set. One might note the similarity to 'improvement heuristics', such as the ones used to find good solutions to traveling salesman problems.

Algorithm 3.5. To solve programs in a family $\mathrm{IP}_{A,\mathbf{c}}$:

```
Input a test set $\mathcal{G}_{\mathbf{c}}$ for the family $\mathrm{IP}_{A,\mathbf{c}}$
    and some $\mathbf{x} \in \mathbb{N}^n$        ($\mathbf{x}$ is feasible for $\mathrm{IP}(A\mathbf{x})$)
Repeat find $\mathbf{g} \in \mathcal{G}_{\mathbf{c}}$ such that $\mathbf{x} - \mathbf{g} \geq \mathbf{0}$,
    $\mathbf{x} \longrightarrow \mathbf{x} - \mathbf{g}$
Until optimal.
```

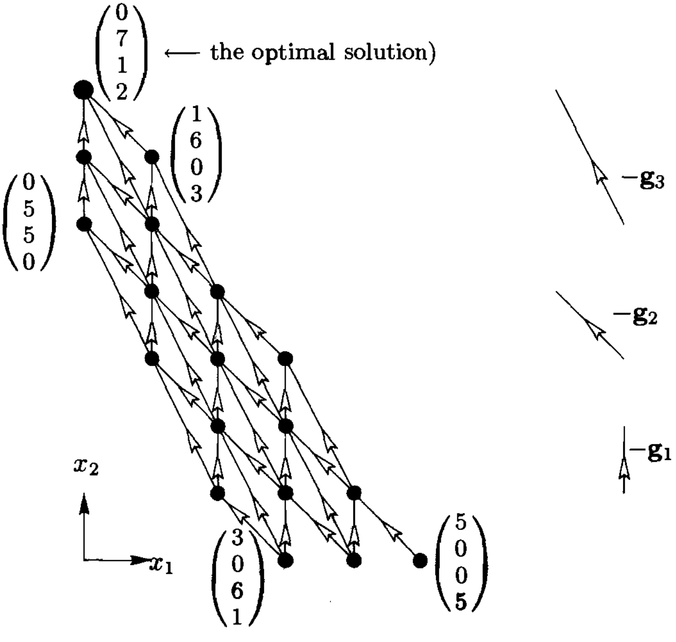
Example 3.6. (Thomas [18]) For the family of integer programming problems that are given by

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \quad \text{and} \quad \mathbf{c}^\top = (1\ 3\ 14\ 17)$$

a minimal test set consists of just three vectors,

$$\mathbf{g}_1 = \begin{pmatrix} 0 \\ -1 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{g}_2 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \quad \mathbf{g}_3 = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 0 \end{pmatrix},$$

and these three are sufficient to do the optimization for an arbitrary right hand side. For $\mathbf{b} = (10\ 15)^\top$, we obtain the situation displayed in the figure, which shows the projection to the (x_1, x_2)-plane: there are 18 feasible points, and the three test set vectors are sufficient (and necessary!) to get you from any feasible point to the optimal one.



Theorem 3.7. (Thomas [18, Cor. 2.1.10]) *The unique minimal test set for the family* $\mathrm{IP}_{A,\mathbf{c}}$ *of integer programs is given by*

$$\begin{aligned}\mathcal{G}_{\mathbf{c}} \quad := \quad \big\{\mathbf{a}_i - \mathbf{a}_i^* \;\big|\; & \mathbf{a}_i \in \min\nolimits_{\preceq}\{\mathbf{x} \in \mathbb{N}^n \textit{ non-optimal}\}, \\ & \mathbf{a}_i^* \textit{ optimal for } \mathrm{IP}(A\mathbf{a}_i^*), \textit{ where } A\mathbf{a}_i = A\mathbf{a}_i^*\big\}.\end{aligned}$$

The connection between Integer Programming and Gröbner Basis Theory can now be made by observing that the minimal test set of Theorem 3.7 corresponds to the reduced Gröbner basis of the binomial ideal

$$I_A \quad := \quad \big\langle \mathbf{X}^{\mathbf{a}^+} - \mathbf{X}^{\mathbf{a}^-} \mid A\mathbf{a} = \mathbf{0},\ \mathbf{a} \in \mathbb{Z}^n \big\rangle$$

with respect to the term order $\prec_{\mathbf{c}}$. (Here $\mathbf{a}^+$ is our shorthand for the vector we obtain by replacing all negative coordinates of $\mathbf{a}$ by zero. Similarly, we use $\mathbf{a}^- := (-\mathbf{a})^+$, so that we have $\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-$, with $\mathbf{a}^+, \mathbf{a}^- \geq \mathbf{0}$. As is customary, $\mathbf{X}^{\mathbf{a}}$ is a notation for $X_1^{a_1} \cdots X_n^{a_n}$, etc.)

The information that $\mathcal{G}_{\mathbf{c}}$ is a Gröbner basis of I_A is not terribly helpful, since in general we do not know a generating set for the ideal — so we can't compute a Gröbner basis, either. Thus we use a small dirty trick: we create a larger integer program, which has an obvious integer feasible point, and for which the ideal has a nice generating set to start from. (Versions of this trick appear both in algebra, see [6, Sect. 3.3], and in linear programming, where slack variables are introduced to obtain 'Phase I' problems that have feasible starting basis, see the 'big-M method' in [14, Sect. 11.2].)

For this, we consider the 'extended integer programs'

$$\mathrm{EIP}(\mathbf{b}) \qquad \min\nolimits_{\prec_{(M\mathbf{1},\mathbf{c})}} \Big\{ \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \in \mathbb{N}^{n+m} \;\Big|\; I_m\mathbf{y} + A\mathbf{x} = \mathbf{b} \Big\},$$

where $M \in \mathbb{N}$ is a large constant, I_m is the $m \times m$ identity matrix, and $\mathbf{1}$ denotes the vector of all ones. We use

$$\mathrm{EIP}_{A,\mathbf{c}}$$

to denote the whole family of these integer programs, with fixed A and $\mathbf{c}$, but varying right hand side $\mathbf{b}$. What have we gained? On the one hand, all of the programs $\mathrm{EIP}(\mathbf{b})$ are feasible: they have the obvious solution $\mathbf{x} = \mathbf{0}$, $\mathbf{y} = \mathbf{b}$. However, an optimal solution will satisfy $\mathbf{y} = \mathbf{0}$, $\mathbf{x} = \mathbf{x}_0$ if the program $\mathrm{IP}(\mathbf{b})$ is feasible, because M was chosen to be *very* large. If $\mathrm{IP}(\mathbf{b})$ is infeasible, then the extended program $\mathrm{EIP}(\mathbf{b})$ has an optimal solution with $\mathbf{y} \neq \mathbf{0}$. The binomial ideal that corresponds to $\mathrm{EIP}_{A,\mathbf{c}}$ *does* have a nice generating set that we can use to start a Buchberger algorithm.

Proposition 3.8. (Conti & Traverso [5]) *The ideal*

$$I_{(I_m,A)} \quad := \quad \Big\langle \mathbf{Y}^{\mathbf{a}_1^+}\mathbf{X}^{\mathbf{a}_2^+} - \mathbf{Y}^{\mathbf{a}_1^-}\mathbf{X}^{\mathbf{a}_2^-} \;\Big|\; (I_m, A)\begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} = \mathbf{0},\ \begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} \in \mathbb{Z}^{m+n} \Big\rangle$$

is generated by the binomials

$$\mathbf{Y}^{A\mathbf{e}_j} - X_j \qquad \textit{for } 1 \le j \le n.$$

The reduced Gröbner basis of $I_{(I_m,A)}$ with respect to the term order $\prec_{(M\mathbf{1},\mathbf{c})}$ yields the minimal test set $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$ for the family $\mathrm{EIP}_{A,\mathbf{c}}$*, via the canonical bijection*

$$\begin{pmatrix}\mathbf{a}_1\\ \mathbf{a}_2\end{pmatrix} \quad\longleftrightarrow\quad \mathbf{Y}^{\mathbf{a}_1^+}\mathbf{X}^{\mathbf{a}_2^+} - \mathbf{Y}^{\mathbf{a}_1^-}\mathbf{X}^{\mathbf{a}_2^-}.$$

The binomials $\mathbf{Y}^{A\mathbf{e}_j} - X_i$ form a Gröbner basis for the ideal that they generate, for a lexicographic term order with $X_i \succ_{\text{lex}} Y_j$. To see that this is the whole ideal $I_{(I_m,A)}$, start with any binomial in $I_{(I_m,A)}$, reduce it to a binomial that contains no X-variables using the generators of $I_{(I_m,A)}$, and then conclude that you have arrived at the zero binomial, using Exercise 6.4.

Putting things together, we have an extremely simple algorithm for integer programming: we 'only' need to compute a reduced Gröbner basis with respect to the term order $\prec_{(M\mathbf{1},\mathbf{c})}$, and then use this with the above algorithm to solve the extended programs $\mathrm{EIP}_{A,\mathbf{c}}$.

Algorithm 3.9 (Integer Programming via Buchberger's Algorithm) The following procedure solves the extended integer program

$$\mathrm{IP}(\mathbf{b}) \qquad\qquad \min\nolimits_{\prec_\mathbf{c}} \{\mathbf{x} \in \mathbb{N}^n | A\mathbf{x} = \mathbf{b}\}$$

for $A \in \mathbb{N}^{m\times n}$, $\mathbf{b} \in \mathbb{N}^m$, $\mathbf{c} \in \mathbb{N}^n$.

First Phase: Compute a test set

Input A, $\mathbf{c}$
Compute the reduced Gröbner basis $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$ for $I := \langle \mathbf{Y}^{A\mathbf{e}_j} - X_j \rangle$,
Output the test set $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$.

Second Phase: Reduction

Input $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$, $\mathbf{b}$
Reduce the monomial $\mathbf{Y}^\mathbf{b}$ with respect to $\mathcal{G}_{(M\mathbf{1},\mathbf{c})}$, get $\mathbf{Y}^{\mathbf{a}_1}\mathbf{X}^{\mathbf{a}_2}$.
Output If $\mathbf{a}_1 \neq \mathbf{0}$, return 'infeasible'.
If $\mathbf{a}_1 = \mathbf{0}$, return '$\mathbf{x}_0 = \mathbf{a}_2$ is optimal'.

While the first phase of this algorithm (computation of a Gröbner basis) amounts to hard work, the second one should typically be quite easy & fast (if we manage to efficiently search the Gröbner basis, which may be huge). But even if we cannot obtain a complete Gröbner basis from the first phase, then we can still use any partial basis to reduce the monomial $\mathbf{Y}^\mathbf{b}$, which may yield a feasible, or even the optimal, point.

However, we are still making quite a detour in Algorithm 3.9: one can formulate the Buchberger algorithm so that it operates directly on lattice points (no ideals, binomials, etc., involved!). This geometric formulation (given in the next section) yields an extremely simple algorithm for integer programming: also one that is very easy to implement! The basic version is not terribly efficient: but we will discuss a few basic ideas about 'how to speed it up'.

4. A Geometric Buchberger Algorithm

The Buchberger algorithm for integer programming is a special case of the general Buchberger algorithm. However, there is a lot of special features in the special situation of 'toric ideals' that we are dealing with here. In particular, one only has to deal with 'binomials with disjoint supports': thus we can get an entirely geometric formulation of the algorithm, dealing with lattice vectors in $\mathbb{Z}^n$ — no polynomials whatsoever appear. This simplifies the data structures considerably!

The translation process may be done as follows. The first observation is that, by definition,

$$I_{(I_m,A)} \quad = \quad \langle \mathbf{Y}^{A\mathbf{e}_j} - X_j \mid 1 \le j \le n \rangle$$

is a *binomial ideal*, an ideal generated by binomials. Any S-pair of two binomials is a binomial (see Exercise 6.3 for a sharper version of this fact). Also the reduction of binomials by binomials leads to binomials. Thus the entire Buchberger process produces only binomials during its lifetime, and any reduced Gröbner basis of $I_{(I_m,A)}$ consists of binomials.

As the second step in our translation process we notice that, whenever a binomial appears in the computation whose two terms have a common factor, we may remove that factor, and the corresponding 'reduced' binomial is also contained in $I_{(I_m,A)}$. This follows from the stronger statement that I is a 'lattice ideal', in the following way.

A *lattice* is a discrete additive subgroup $\mathcal{L} \subseteq \mathbb{Z}^n$, that is, the set of all integral linear combinations of a finite set of linearly independent vectors in $\mathbb{R}^n$. With every lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ we associate the *lattice ideal*

$$I_{\mathcal{L}} \quad := \quad \langle \mathbf{X}^{\mathbf{a}^+} - \mathbf{X}^{\mathbf{a}^-} \mid \mathbf{a} \in \mathcal{L} \rangle.$$

Thus, Proposition 3.8 shows that $I_{(I_m,A)}$ is a lattice ideal. Note that in the definition of $I_{\mathcal{L}}$, we can replace $\mathcal{L}$ by the subset of all lattice vectors that are positive with respect to the ordering $\succ$ that we consider, that is, by

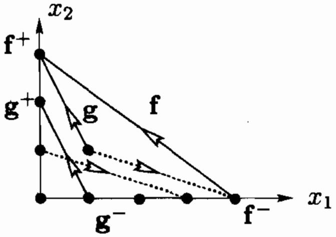
$$\mathcal{L}^{\succ \mathbf{0}} \quad := \quad \{\mathbf{a} \in \mathcal{L} \mid \mathbf{a} \succ \mathbf{0}\}.$$

Thus the Buchberger algorithm can immediately remove the common factors from all binomials that it produces. (In particular, any reduced Gröbner basis of I contains only binomials of the form $\mathbf{X}^{\mathbf{a}^+} - \mathbf{X}^{\mathbf{a}^-}$ with $\mathbf{a} \in \mathcal{L}^{\succ \mathbf{0}}$.) Thus the Buchberger algorithm can really be formulated as a geometric algorithm operating on lattice vectors. So we get the following two algorithms to compute the *reduced Gröbner basis of a lattice*, that is, the finite subset $\mathcal{G} \subseteq \mathcal{L}^{\succ \mathbf{0}}$ that corresponds to the reduced Gröbner basis of $I_{\mathcal{L}}$. The assumption for this is that we know a 'good' generating set for the lattice, i.e., a subset of the lattice corresponding to a set of binomials that generates $I_{\mathcal{L}}$.

Algorithm 4.1 (Reduction). The following algorithm computes the reduction of a vector $\mathbf{f} \in \mathbb{Z}^{m+n}$ by a set $\mathcal{G}$ of integer vectors. Compare it to the algorithm *Reduce* of Chapter 1!

```
**Input** $\mathcal{G} \subseteq \mathcal{L}^{\succ 0}$, $\mathbf{f} \succ \mathbf{0}$.
**Repeat**
  If there is some $\mathbf{g} \in \mathcal{G}$ with $\mathbf{g}^+ \leq \mathbf{f}^+$, then replace $\mathbf{f}$ by $\pm(\mathbf{f}-\mathbf{g}) \succ \mathbf{0}$.
  If there is some $\mathbf{g} \in \mathcal{G}$ with $\mathbf{g}^+ \leq \mathbf{f}^-$, then replace $\mathbf{f}$ by $\mathbf{f}+\mathbf{g}$.
**Output** $\overline{\mathbf{f}} := \mathbf{f}$.
```

Our figure illustrates the first case in the reduction algorithm, where we have $\mathbf{g}^+ \leq \mathbf{f}^+$, and the reduced vector arises as a difference. (Lattice vectors such as $\mathbf{g}$ can be drawn with the head at $\mathbf{g}^+$ and the tail at $\mathbf{g}^-$.)



You should check that this reduction process corresponds to the reduction of $\mathbf{X}^{\mathbf{f}^+} - \mathbf{X}^{\mathbf{f}^-} = X_2^3 - X_1^4$ by $\mathbf{X}^{\mathbf{g}^+} - \mathbf{X}^{\mathbf{g}^-} = X_2^2 - X_1$, where the resulting polynomial $X_1^4 - X_1X_2$ has a common factor X_1, whose removal corresponds to a translation of the dotted vector.

Algorithm 4.2 (Buchberger Algorithm on Lattice Vectors). The following algorithm computes the reduced Gröbner basis of the lattice $\mathcal{L}$, for a fixed term order $\succ$.

First Step: Construct a Gröbner basis

```
**Input** A basis $\{\mathbf{a}_1, \dots, \mathbf{a}_n\} \subseteq \mathcal{L}$ of the lattice $\mathcal{L}$ such that the binomials
$\mathbf{X}^{\mathbf{a}^+} - \mathbf{X}^{\mathbf{a}^-}$ generate $I_{\mathcal{L}}$.          (See Exercise 6.5!)
**Set** $\mathcal{G}_{old} := \emptyset$, $\mathcal{G} := \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$
**Repeat** While $\mathcal{G}_{old} \neq \mathcal{G}$, repeat the following steps
  $\mathcal{G}_{old} := \mathcal{G}$
  ($S$-pairs) construct the pairs $\mathbf{g} := \mathbf{a} - \mathbf{a}' \succ \mathbf{0}$ with $\mathbf{a}, \mathbf{a}' \in \mathcal{G}$.
  (Reduction) reduce the vectors $\mathbf{g}$ by the vectors in $\mathcal{G}_{old}$. If $\overline{\mathbf{g}} \neq \mathbf{0}$, set
  $\mathcal{G} := \mathcal{G} \cup \overline{\mathbf{g}}$.
```

Second Step: Construct a minimal Gröbner basis

```
**Repeat** If for some $\mathbf{g} \in \mathcal{G}$ the point $\mathbf{g}^+$ can be reduced by some $\mathbf{g}' \in \mathcal{G} \backslash \mathbf{g}$,
then delete $\mathbf{g}$ from $\mathcal{G}$.
```

Third Step: Construct the reduced Gröbner basis

```
**Repeat** If for some $\mathbf{g} \in \mathcal{G}$ the point $\mathbf{g}^-$ can be reduced by some $\mathbf{g}' \in \mathcal{G} \backslash \mathbf{g}$,
then replace $\mathbf{g}$ by the corresponding reduced vector: $\mathcal{G} := \mathcal{G} \backslash \mathbf{g} \cup \overline{\mathbf{g}}$.
**Output** $\mathcal{G}_{red} := \mathcal{G}$.
```

All these operations are easy to visualize (at least in the 2-dimensional situation). They are also easily implemented — just do it. There is also a lot of flexibility: in fact, for a successful implementation it is important to reduce earlier, otherwise the Gröbner bases constructed in the 'First Step' will be too large. See [5, 12, 21] for further ideas about how to make this efficient.

We just remark that the elements that can occur in a reduced Gröbner basis can be characterized geometrically in a different way. The following theorem is due to Sturmfels & Thomas [16].

Theorem 4.3. *The universal Gröbner basis (that is, the union of all the reduced Gröbner bases $\mathcal{G}_{\mathbf{c}}$ of* $\mathrm{IP}_{A,\mathbf{c}}$*, for all objective functions* $\mathbf{c}$*) consists of all the primitive lattice vectors* $\mathbf{a} \in \mathbb{Z}^n$ *(with* $A\mathbf{a} = \mathbf{0}$*, and* $\frac{1}{\lambda}\mathbf{a} \notin \mathbb{Z}^n$ *for* $\lambda > 1$*) such that* $[\mathbf{a}^+, \mathbf{a}^-]$ *is an edge of the polyhedron*

$$P_I(\mathbf{a}^+) \quad = \quad \mathrm{conv}\{\mathbf{x} \in \mathbb{N}^n \mid A\mathbf{x} = A\mathbf{a}^+\}.$$

This theorem can be applied as well to the extended integer programs EIP_A, where we know how the minimal test sets (Gröbner bases) can be computed via Buchberger's algorithm. Sturmfels & Thomas [16] also have a technique to compute universal Gröbner bases via one single application of a Buchberger algorithm (to a larger problem).

5. A Variant of the Buchberger Algorithm

The following presents a variation of the Buchberger algorithm that may be even more useful for integer programming.

Given a matrix $A \in \mathbb{N}^{m \times n}$, an objective function $\mathbf{c} \in \mathbb{N}^n$, and a right hand side vector $\mathbf{b} \in \mathbb{N}^m$, we denote by $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$ the optimization problem

$$\max_{\prec_{\mathbf{c}}} \{\mathbf{x} \in \mathbb{N}^n \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}.$$

This is a special but quite common type of integer program, which we call a problem in *inequality standard form with upper bounds.* See also Exercise 6.1. Again, in order to avoid dealing with degenerate cases we refine the objective function $\mathbf{c}^\top\mathbf{x}$ to get a term order $\prec_{\mathbf{c}}$. A *test set* for a problem of the type $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$ is a set $\mathcal{G}$ of vectors $\mathbf{g} \succ_{\mathbf{c}} \mathbf{0}$ such that every non-optimal feasible point can be improved by one of the test set vectors. Algebraically, both $\mathbf{u}$ and $\mathbf{b}$ provide 'degree bounds' for Buchberger algorithms. Thus test sets for families of problems of the type $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$ correspond to certain truncated Gröbner bases. However, our discussion in the following stays in the elementary geometry setting of [21]; The algebraic picture can be found in [20].

Roughly speaking, a test set, $\mathcal{G}$ say, can be computed as follows. Start with the n unit vectors, i.e., set $\mathcal{G} := \{e_i \mid 1 \leq i \leq n\}$. Iteratively, compute the difference vectors between all pairs of vectors that are in $\mathcal{G}$ and direct each such difference vector such that it is greater than $\mathbf{0}$ with respect to the order. All such difference vectors are added to $\mathcal{G}$, if they are not already in $\mathcal{G}$, and if they are differences of feasible points for $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$. The algorithm terminates when no more vectors are added to $\mathcal{G}$.

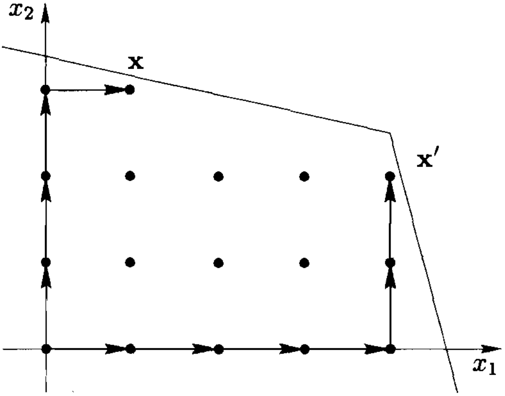
More precisely, the basic algorithm can be formulated as follows:

Algorithm 5.1. To compute a test set for integer programs $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$ in inequality standard form with upper bounds.

```
  Input $A$ and $\prec_{\mathbf{c}}$
  Initialize Set $\mathcal{G}_{old} := \emptyset$,  $\mathcal{G} := \{e_i \mid 1 \le i \le n\}$.
  While $\mathcal{G}_{old} \neq \mathcal{G}$ perform the following steps:
    Set $\mathcal{G}_{old} := \mathcal{G}$.
    For all pairs of vectors $\mathbf{v}, \mathbf{w} \in \mathcal{G}$ such that
        $\mathbf{w} \succ_{\mathbf{c}} \mathbf{v}$,  $-\mathbf{b} \le A(\mathbf{w} - \mathbf{v}) \le \mathbf{b}$  and  $-\mathbf{u} \le \mathbf{w} - \mathbf{v} \le \mathbf{u}$,
    set $\mathcal{G} := \mathcal{G} \cup \{\mathbf{w} - \mathbf{v}\}$.
```

Whenever the loop in this algorithm is executed (except for the last time), a new vector is added to the set $\mathcal{G}_{old}$. Since the number of integral vectors $\mathbf{x}$ satisfying $-\mathbf{u} \le \mathbf{x} \le \mathbf{u}$ is bounded by $\prod_{i=1}^{n}(2u_i + 1)$, the above algorithm terminates after finitely many steps.

Let us now show that the set $\mathcal{G}$ generated by the above algorithm is a test set for $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$. Suppose that $\mathbf{x}$ is a feasible point ($A\mathbf{x} \le \mathbf{b}$, $\mathbf{0} \le \mathbf{x} \le \mathbf{u}$) that cannot be improved by any element in $\mathcal{G}$, and let $\mathbf{x}'$ be a feasible vector with $\mathbf{x}' \succ_{\mathbf{c}} \mathbf{x}$. Then $\mathbf{x}' - \mathbf{x}$ is not an element of $\mathcal{G}$. However, as $\mathbf{x}' - \mathbf{x}$ can be written as a linear combination of unit vectors and since unit vectors are elements of $\mathcal{G}$, we can decrease from $\mathbf{x}$ to reach $\mathbf{0}$, then increase to reach $\mathbf{x}'$.



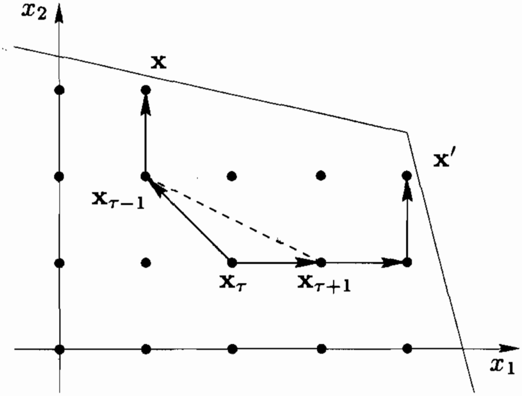
Hence, there exists a sequence $P = (\mathbf{x}^0, \ldots, \mathbf{x}^p)$ of vectors $\mathbf{x}^i$ and a number $1 \le \tau < p$ with the properties:

(i) $\mathbf{x}^0 = \mathbf{x}$, $\mathbf{x}^p = \mathbf{x}'$,
(ii) for all $i = 1, \ldots, \tau$, $-(\mathbf{x}^i - \mathbf{x}^{i-1}) \in \mathcal{G}$,
(iii) for all $i = \tau + 1, \ldots, p$, $(\mathbf{x}^i - \mathbf{x}^{i-1}) \in \mathcal{G}$,
(iv) every vector in P is feasible.

Let $p_{\min}$ be the smallest number such that there exists some sequence

$$P_{\min} \quad = \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{p_{\min}})$$

of vectors $\mathbf{x}^i$ and a number $1 \leq \tau < p_{\min}$ satisfying (i), (ii), (iii) and (iv).



Since $-(\mathbf{x}^\tau - \mathbf{x}^{\tau-1}) \in \mathcal{G}$ and $(\mathbf{x}^{\tau+1} - \mathbf{x}^\tau) \in \mathcal{G}$, the difference vector

$$\mathbf{v} := (\mathbf{x}^{\tau+1} - \mathbf{x}^\tau) \; - \; (-(\mathbf{x}^\tau - \mathbf{x}^{\tau-1})) \quad = \quad \mathbf{x}^{\tau+1} \; - \; \mathbf{x}^{\tau-1}$$

has been computed in the while loop of Algorithm 5.1. Moreover, both vectors $\mathbf{x}^{\tau+1}$ and $\mathbf{x}^{\tau-1}$ are feasible. It follows that $-\mathbf{b} \leq A\mathbf{v} \leq \mathbf{b}$ and $-u_i \leq v_i \leq u_i$ for all i.

In case that $\mathbf{0} \prec_\mathbf{c} \mathbf{v}$, the vector $\mathbf{v}$ was added to $\mathcal{G}$. Consequently,

$$P' \quad := \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{\tau-1}, \mathbf{x}^{\tau+1}, \ldots, \mathbf{x}^{p_{\min}})$$

and $\tau - 1$ again satisfy properties (i)–(iv), yet involving $p_{\min} - 1$ vectors, a contradiction.

Therefore $\mathbf{v} \prec_\mathbf{c} \mathbf{0}$. In this case the vector $-\mathbf{v}$ was added to $\mathcal{G}$ in Algorithm 5.1. Then,

$$P' \quad := \quad (\mathbf{x}^0, \ldots, \mathbf{x}^{\tau-1}, \mathbf{x}^{\tau+1}, \ldots, \mathbf{x}^{p_{\min}})$$

and τ satisfy properties (i)–(iv). Since again only $p_{\min} - 1$ vectors belong to P', we obtain a contradiction.

Thus we have proved the following theorem.

Theorem 5.2. *Algorithm 5.1 terminates after a finite number of steps. The output is a test set for the integer programming problem* $\mathrm{IP}_{A,\mathbf{b},\mathbf{c}}$.

Compared to Algorithm 3.9, this extremely simple algorithm has some essential advantages. In particular, it works without the increase in dimension to obtain the extended problem: the computation takes place in the original space. However, Algorithm 5.1 still has the problem that it computes too many elements: the partial Gröbner basis computed is way too large. (For the basic version of the algorithm presented here, nearly all difference vectors of feasible points will be contained in $\mathcal{G}$!) Thus one has to work with reduction, and thus discard superfluous elements during the computation (see [21]). Also, this algorithm sometimes makes way too many comparisons, while generating only relatively few new basis elements. Such observations, made on a practical implementation, led to further variations of the algorithm that are currently still under investigation.

6. Exercises

Exercise 6.1 (Standard Forms of Integer Programs). Show that the 'equality standard form'

$$\min\{\mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

and the 'inequality standard form'

$$\max\{\mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\}$$

of linear programs are equivalent: for any problem in one form we can construct a problem in the other form that solves it.
(Assume that $A \in \mathbb{N}^{m\times n}$ has no zero columns, $\mathbf{b} \in \mathbb{N}^m$, and $\mathbf{c} \in \mathbb{Z}^n$.)

Exercise 6.2 (Upper Bounds). For a problem of the form

$$\max\{\mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\},$$

with $A \in \mathbb{N}^{m\times n}$, $\mathbf{b} \in \mathbb{N}^m$, and $\mathbf{c} \in \mathbb{Z}^n$, how can we compute upper bounds u_i for the variables x_i? What happens in the special case when A has a zero column?

The following problem sharpens our observation that S-pair formation corresponds to difference of vectors: we explicitly identify the 'superfluous' monomial factors that occur in the formation of S-pairs.

Exercise 6.3 (S-Pairs and Difference Vectors). For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$, $\mathbf{a}, \mathbf{b} \succ \mathbf{0}$, the S-polynomial of $\mathbf{X}^{\mathbf{a}^+} - \mathbf{X}^{\mathbf{a}^-} \in I_{\mathcal{L}}$ and $\mathbf{X}^{\mathbf{b}^+} - \mathbf{X}^{\mathbf{b}^-} \in I_{\mathcal{L}}$ is

$$\mathbf{X}^{\min(\mathbf{a}^+,\mathbf{b}^+)}\,(\mathbf{X}^{(\mathbf{a}-\mathbf{b})^+} - \mathbf{X}^{(\mathbf{a}-\mathbf{b})^-}),$$

a monomial times the binomial corresponding to $\mathbf{a} - \mathbf{b}$.

Exercise 6.4 (Binomial Criterion). A binomial $\mathbf{X}^{\mathbf{a}} - \mathbf{X}^{\mathbf{b}}$, with $\mathbf{a}, \mathbf{b} \geq \mathbf{0}$, is contained in $I_{\mathcal{L}}$ if and only if $\mathbf{a} - \mathbf{b} \in \mathcal{L}$.
(Hint: $\mathbf{X}^{\mathbf{a}}$ and $\mathbf{X}^{\mathbf{b}}$ reduce to the same standard monomial.)

Exercise 6.5 (Ideal of a Lattice: Generators). Assume that the lattice $\mathcal{L}$ is generated by the columns of a nonnegative matrix $A \in \mathbb{N}^{n \times n}$. Show that then the ideal $I_{\mathcal{L}}$ is generated by the binomials

$$\mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-}.$$

Show that this can fail if we do not assume A to be nonnegative. (A more general version of this is [21, Lemma 2.1].)

Exercise 6.6 (Gröbner Bases of a Lattice: an Example). Let $\mathcal{L}_A \subseteq \mathbb{Z}$ be the 2-dimensional lattice generated by the columns of

$$A = \begin{pmatrix} 1 & 4 \\ 4 & 3 \end{pmatrix}.$$

Compute all the (four) different reduced Gröbner bases for the corresponding ideal. Describe the structure of the various Gröbner basis elements. How many standard monomials are there in each case?

Exercise 6.7 (Gröbner Bases of a Lattice: Geometry). Show that if $\mathcal{L}$ is a 2-dimensional integral lattice, then the universal Gröbner basis (the union of all the reduced Gröbner bases) consists of the following lattice vectors:

- the vertices $\mathbf{a} \in \mathbb{Z}^2$ of the polyhedron $\operatorname{conv}(\mathcal{L} \cap \mathbb{N}^2 \backslash \mathbf{0})$, and
- the vectors $\mathbf{a} \in \mathcal{L}$ that have one positive and one negative component, and for which $\mathbf{0}$ and $\mathbf{a}$ are two adjacent vertices of the polyhedron

$$\operatorname{conv}(\mathcal{L} \cap \{\mathbf{x} \in \mathbb{Z}^2 \mid \mathbf{x} \geq -\mathbf{a}^-\}).$$

(Remark: a similar structure theorem is true in higher dimensions as well, but harder to prove [17].)

Exercise 6.8 (A Variant of Buchberger's Algorithm: an Example). Apply Algorithm 5.1 to compute a test set $\mathcal{G}$ for the 0/1 knapsack problem

$$\max\{x_1 + 2x_2 + 3x_3 \mid x_1 + 2x_2 + 3x_3 \leq 3,\ x_i \in \{0,1\},\ i = 1,2,3\}.$$

Identify a minimal test set $\mathcal{G}_{\min} \subseteq \mathcal{G}$.

Notes

While the theory of Gröbner bases [1, 2, 3, 6] yields basic ideas and tools, the discussion in this chapter stays in an 'elementary geometry' setting. Chvátal [4] and Schrijver [14] are excellent guides to all topics related to Linear and Integer Programming. [22] is a recent exposition of the geometry and combinatorics of polytopes.

As mentioned in the introduction, the basic ideas of Section 3 are due to Conti & Traverso [5]. The connection between lattices, binomial ideals and Gröbner bases is relevant to interesting aspects in the theory of integer programming (the 'local situation', Gomory's [9] 'group problem'), but also, for example, to the ideals of toric varieties. The key reference for these directions is Sturmfels [15]. See [8] for more on binomial ideals.

Our presentation in Sections 3 and 4 is based on Thomas [18] [19, Chap. 2]. I am very grateful to Rekha Thomas for many helpful comments and discussions on this chapter, and for her permission to report about and draw on her materials.

The ideas for Section 5 are from [21]. An algebraic interpretation of the situation in terms of 'truncated Gröbner bases' was given in [20].

We refer to [10] for an alternative approach to the 'phase I' problem. Recently, Li, Guo, Ida, Darlington [11] have described a combination of truncated Gröbner bases with the Hoşten-Sturmfels approach. They also reported some computational tests: on random problems of sizes up to 8 × 16 —which must still be considered very modest for all practical purposes. Computational results (also for larger, structured problems) are also presented in [10], in [7], and in [21].

A successful application of the Buchberger approach to a class of integer programming problems arising in practice was reported in Natraj, Tayur & Thomas [13].

References

1. W. W. Adams and P. Loustaunau (1994): *An Introduction to Gröbner Bases*, Graduate Studies in Math., Vol. III, American Math. Soc., Providence RI.
2. T. Becker and V. Weispfennig (1993): *Gröbner Bases: A Computational Approach to Commutative Algebra*, Graduate Texts in Mathematics **141**, Springer-Verlag, New York Berlin Heidelberg.
3. B. Buchberger (1985): *Gröbner bases: An algorithmic method in polynomial ideal theory*, in: N.K. Bose (ed.), 'Multidimensional Systems Theory', D. Reidel, 184–232.
4. V. Chvátal (1983): *Linear Programming*, Freeman, New York.
5. P. Conti and C. Traverso (1991): *Buchberger algorithm and integer programming*, pp. 130–139 in Proceedings AAECC-9 (New Orleans), Springer, Lecture Notes in Computer Science **539**.
6. D. A. Cox, J. B. Little, and D. O'Shea (1992): *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Undergraduate Texts in Mathematics, Springer-Verlag, New York Berlin Heidelberg.
7. F. Di Biase and R. Urbanke (1995): *An algorithm to calculate the kernel of certain polynomial ring homomorphisms*, Experimental Math. **4**, 227–234.
8. D. Eisenbud and B. Sturmfels (1996): *Binomial ideals*, Duke Math. J. **84**, 1–45.
9. R. E. Gomory (1969): *Some polyhedra related to combinatorial problems*, Linear Algebra and its Applications **2**, 451–455.
10. S. Hoşten and B. Sturmfels (1995): GRIN: *An implementation of Gröbner bases for integer programming*, pp. 267–276 in: "Integer Programming and Combinatorial Optimization" (E. Balas, J. Clausen, eds.), Proc. 4th Int. IPCO Conference (Copenhagen, May), Lecture Notes in Computer Science **920**, Springer-Verlag, Berlin Heidelberg New York.
11. Q. Li, Y. Guo, T. Ida, and J. Darlington (1997): *The minimised geometric Buchberger algorithm: An optimal algebraic algorithm for integer programming*, pp. 331–338 in: Proc. ISSAC'97, ACM Press.
12. C. Moulinet and L. Pottier (1997): *Gröbner bases of toric ideals: properties, algorithms, and applications*, preprint, INRIA Sophia Antipolis, 10 pages.

13. N.R. Natraj, S.R. Tayur, and R.R. Thomas (1995): *An algebraic geometry algorithm for scheduling in presence of setups and correlated demands*, Math. Programming **69A**, 369–401.
14. A. Schrijver (1986): *Theory of Linear and Integer Programming*, Wiley-Interscience, Chichester.
15. B. Sturmfels (1995): *Gröbner Bases and Convex Polytopes*, AMS University Lecture Series, Vol. 8, American Math. Soc., Providence RI.
16. B. Sturmfels and R.R. Thomas (1997): *Variation of cost functions in integer programming*, Math. Programming **77**, 357–387.
17. B. Sturmfels, R. Weismantel, and G.M. Ziegler (1995): *Gröbner bases of lattices, corner polyhedra, and integer programming*, Beiträge Algebra und Geometrie/Contributions to Algebra and Geometry **36**, 281–298.
18. R.R. Thomas (1995): *A geometric Buchberger algorithm for integer programming*, Math. Operations Research **20**, 864–884.
19. R.R. Thomas (1994): *Gröbner basis methods for integer programming*, Ph. D. Thesis, Cornell University, 157 pages.
20. R.R. Thomas and R. Weismantel (1997): *Truncated Gröbner bases for integer programming*, Applicable Algebra in Engineering, Communication and Computing (AAIECC), **8**, 241–257.
21. R. Urbaniak, R. Weismantel, and G.M. Ziegler (1997): *A variant of Buchberger's algorithm for integer programming*, SIAM J. Discrete Math. **10**, 96–108.
22. G.M. Ziegler (1995): *Lectures on Polytopes*, Graduate Texts in Mathematics **152**, Springer-Verlag, New York Berlin Heidelberg.

Chapter 8. Working with Finite Groups

Hans Cuypers, Leonard H. Soicher, and Hans Sterk

1. Introduction

Two common ways to describe groups are to present them by generators and relations or as automorphism groups of algebraic, geometric or combinatorial structures. (Think of linear groups acting on vector spaces, symmetry groups of regular polytopes, Galois groups etc.) An automorphism group of such a structure may also be considered to be a subgroup of the group of all permutations of the elements of that structure. Automorphism groups can thus be seen as permutation groups. Permutation groups are groups consisting of permutations of a set with composition of permutations as group multiplication. So, for example, we may view linear groups as permutation groups on the set of vectors of the underlying vector space (but this may not be the most efficient approach). The Todd-Coxeter coset enumeration method provides, among other things, a link between groups given by generators and relations on the one hand and permutation groups on the other.

Since permutations of a finite set (say $\{1, 2, \ldots, n\}$) can be easily dealt with on a computer, this opens up the way for computations in groups. Permutation group algorithms is now the most developed area of Computational Group Theory, and is still being actively developed (see for example [26, 27, 14, 15, 8, 9]).

In Section 2, we describe some of the basic permutation group algorithms. We describe algorithms for computing the order of a group, testing nilpotency or solvability, and algorithms for computing generators for particular subgroups like centralizers, normalizers, and stabilizers of sets or elements. Since we are dealing with permutation groups acting faithfully on a set Ω of size n say, such a group has order at most $n!$. Thus, it is certainly possible to perform all kinds of computations within this group in a finite amount of time. However, we will mainly concentrate on algorithms with running time polynomial in n.

By way of illustration, in Project 6, we make use of some of the algorithms in Section 2 to help with the construction of the small Mathieu groups and some of their interesting subgroups. Coset enumeration is then used as a tool to solve a problem in graph theory: classify all connected graphs which are locally isomorphic to the incidence graph of the biplane of order 11, and admit an ordered-triangle-transitive automorphism group.

Good references for most of the results discussed are [4] for permutation group algorithms and [22, 28] for coset enumeration methods. For actual computations with permutation groups one may use either a general purpose computer algebra system like Maple [6] (if the computations are quite straightforward), or a more specialized system like MAGMA [3] or GAP [24].

2. Permutation Groups

2.1 The Setting

2.1. For a set Ω, we denote by $\mathrm{Sym}(\Omega)$ the *symmetric group* of all permutations (i.e., bijections) of Ω. If $\Omega = \{1, \dots, n\}$, we usually write S_n instead of $\mathrm{Sym}(\Omega)$. Group multiplication is the composition of permutations, but read from left to right in accordance with the implementation in several computer algebra systems.

The basic notion that connects groups to permutations is that of a permutation representation: for a group G, a *permutation representation* of G is a homomorphism of G into the group $\mathrm{Sym}(\Omega)$ for some set Ω. In particular, each element $g \in G$ acts on Ω, that is, produces a permutation of Ω; we denote the image of $\omega \in \Omega$ under the action of g by ω^g. (This means that we let elements of $\mathrm{Sym}(\Omega)$ act on the right.) A *permutation group* is a subgroup of $\mathrm{Sym}(\Omega)$ (this corresponds to the special case where the homomorphism is an inclusion). In this setting the usual terminology is that G *acts* on Ω. In this section we deal with finite groups G and finite sets Ω (the most obvious one: $\{1, 2, \dots, n\}$). For a permutation representation with finite set Ω, the cardinality $|\Omega|$ is called the *degree* of the representation.

Interesting situations usually arise when the sets carry some additional structure. A simple example illustrating this is the group of symmetries of (the graph on the vertices of) a cube, described as a permutation group of the eight vertices. For example, with the labeling of Figure 2.1, the permutation $(1, 2, 3, 4)(5, 6, 7, 8)$ describes a rotation over 90^o, whereas $(2, 5)(3, 8)$ is a reflection in the plane through 1, 6, 7 and 4. (The permutations are written as products of disjoint cycles.)

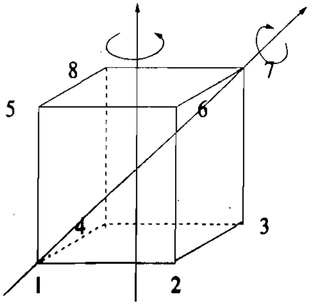


Fig. 2.1. Symmetries of the cube as permutations.

2.2. Producing meaningful permutation representations of a given group can be quite hard. The construction used in the proof of the classical result that every group is isomorphic to a permutation group usually provides little insight: as a set take $\Omega = G$, the group itself, and assign to each $g \in G$ the permutation $R_g : G \to G$, sending $h \in G$ to hg (right multiplication by g). The map $G \to \mathrm{Sym}(G)$, $g \mapsto R_g$ is the required injective homomorphism. An indication of the restricted practical value of this realization is the relatively large size of $\mathrm{Sym}(G)$ and length of its elements (written as permutations) compared to the size of G.

There are several ways of constructing new permutation representations out of a given one, which may be of help in studying specific aspects of the group. For example, the group of symmetries of the cube also acts on its four main diagonals (each encoded as a pair of opposite vertices). This induced action makes it clear for instance that the group of symmetries admits a surjective morphism to the group S_4. Another example is the induced action of a group (acting on the set Ω) on the set of subsets of size k of Ω or the action by multiplication from the right on the right cosets of G with respect to a subgroup (see below).

2.3. To further analyse permutation representations, the following basic notions are also needed.

Let G act on Ω. The *G-orbit* (or orbit for short) of an element $\omega \in \Omega$ is the set

$$\omega^G = \{\omega^g \mid g \in G\}.$$

Orbits evidently partition the set Ω, that is, being in the same orbit defines an equivalence relation on Ω. The group G is said to act *transitively* if Ω itself is an orbit. Equivalently: for every $x, y \in \Omega$ there exists a $g \in G$ with $x^g = y$. A generalization is the notion of a *t–transitive* group: for every two t–tuples $(x_1, x_2, \ldots, x_t)$ and $(y_1, y_2, \ldots, y_t)$, each consisting of t distinct elements of Ω, there exists a $g \in G$ with $x_i^g = y_i$ $(i = 1, \ldots, t)$. A 1–transitive group is called transitive in accordance with the previous definition.

For example, the symmetric group S_n, acting on $\{1, 2, \ldots, n\}$ is clearly n–transitive; the *alternating* group A_n (i.e., the group of even permutations) is $(n-2)$–transitive, but is not n–transitive for $n \geq 2$. The symmetry group of the cube is transitive, but not 2–transitive (two adjacent vertices cannot be transformed into two non–adjacent vertices).

The *stabilizer* of an element ω is the subgroup

$$G_\omega = \{g \in G \mid \omega^g = \omega\}.$$

There is a relation between the cardinalities of G, G_ω and the orbit ω^G of ω, which will be exploited in the following sections, and which we now explain. Define the map

$$f : G \to \omega^G, \quad g \mapsto \omega^g.$$

By the definition of orbit, this map is surjective. Also, if $g, h \in G$, then $f(g) = f(h)$ if and only if $\omega^{hg^{-1}} = \omega$, that is, $hg^{-1} \in G_\omega$. So $f(g) = f(h)$ if and only if the cosets $f^{-1}(f(g)) = G_\omega g = \{xg \mid x \in G_\omega\}$ and $f^{-1}(f(h)) = G_\omega h$ coincide. We conclude that for every $\omega' \in \omega^G$, the set $f^{-1}(\omega')$ has $|G_\omega|$ elements. This proves:

Proposition 2.4. *If a finite group G acts on Ω and $\omega \in \Omega$, then*

$$|G|/|G_\omega| = |\omega^G|.$$

In particular, if G acts transitively, then $|G|/|G_\omega| = |\Omega|$.

2.5. If G acts transitively, then the map constructed in the proof of Proposition 2.4 is a bijection between the set Ω and the set $G_\omega\backslash G$ of right cosets of G_ω in G. Through this bijection, G acts on $G_\omega\backslash G$. The action of an element $g \in G$ on Ω is transferred into right multiplication by g on the elements in $G_\omega\backslash G$:

$$g : G_\omega h \mapsto G_\omega hg.$$

In fact, in a similar way, every subgroup H of G gives rise to a transitive permutation representation of G. The group G acts by right multiplication on the right cosets of H in G. More precisely, for every $g \in G$ the map 'right multiplication by g'

$$R_g : H\backslash G \to H\backslash G, \quad R_g(Hh) = Hhg \text{ for } Hh \in H\backslash G$$

is a (well-defined) bijective map. Then $g \mapsto R_g$ defines a morphism of G into $\mathrm{Sym}(H\backslash G)$. This is a slight variation of the construction used to prove that every group is isomorphic to a permutation group (the case where H is trivial). It enables one to focus on properties of the group G related to the subgroup H.

Since the stabilizer of $H \in H\backslash G$ is H itself, Proposition 2.4 is just another version of

Lagrange's Theorem 2.6. *Let G be a finite group and let H be a subgroup of G. Then*

$$|H| \textit{ divides } |G|.$$

In particular, the order of any element of G (i.e., the order of the group generated by that element) divides $|G|$.

2.2 Computing Orbits and Stabilizers

2.7. A group G is *generated* by its subset X if every element of G can be written as a product $g_1 g_2 \cdots g_m$ where $g_i \in X$ or $g_i^{-1} \in X$ for all i (of course, if the group is finite, then we do not need to include the inverses). The elements of X are called *generators* and we write $G = \langle X \rangle$ to denote that G is generated by X.

In this section we develop the basic tools for computations in groups generated by a set of permutations in S_n. On a computer the generating permutations are stored instead of all of G. As a first example we explain a naive approach to computing the order of such a permutation group. Throughout this section, all permutation groups and the sets on which they act are assumed to be finite.

2.8. The Order of a Group. First compute the order of an orbit of an element ω of $\Omega = \{1, 2, \ldots, n\}$ and then compute the stabilizer of that element. According to Proposition 2.4, the product of the orders of the orbit and the stabilizer equals the order of the group G. To determine the order of the stabilizer, we consider the action of this stabilizer on the set $\Omega \setminus \{\omega\}$ and repeat the process. To make this strategy work, we need algorithms to compute orbits and (generators for) stabilizers.

Example 2.9. In the example of the cube, it is easy to see that the symmetry group G is transitive on the eight vertices of the cube. So the relation $|G| = 8 \cdot |G_1|$ holds. Using the rotation $(2,5,4)(3,6,8)$ we find that the G_1–orbit of 2 contains at least 2, 5 and 4. Since these vertices are the only ones at distance 1 from the vertex 1, this orbit has exactly 3 elements. So we get $|G| = 8 \cdot 3 \cdot |G_{1,2}|$, where $G_{1,2}$ is short for $(G_1)_2$. A similar argument shows that the $G_{1,2}$–orbit of 3 contains 2 elements: 3 and 6 (use the reflection $(4,5)(3,6)$). The stabilizer $G_{1,2,3}$ is trivial so that the order of G is equal to $8 \cdot 3 \cdot 2 = 48$.

2.10. Orbits. The first task is to determine the orbits of a permutation group $G = \langle X \rangle$ generated by the subset X. To find the orbit containing the element ω, here is what you do:

1. Start with the set $\{\omega\}$. This is the initialization of 'orbit-to-be', the set that is to become the full orbit of ω.
2. Have each element of X act on ω. If this doesn't produce any new elements, you are done. Else, put the elements different from ω in a set 'new'.
3. Update 'orbit-to-be' by setting it equal to the union of 'orbit-to-be' and the set 'new'.
4. Have each element of X act on each element of 'new' (rather than of 'orbit-to-be' as this saves work). If this doesn't produce any element not already in 'orbit-to-be', then you're done. Else update 'new' by setting it to contain precisely the elements found at this stage that are not already in 'orbit-to-be'.
5. Go back to 3.

It is useful to store certain information about the action of the elements of X on the orbit elements in a structure called a Schreier tree, defined below.

Definition 2.11. Let $G \leq \mathrm{Sym}(\Omega)$ be generated by a set X, and let $\alpha \in \Omega$. A *Schreier tree* with root α for X is a tree (i.e., graph without cycles), rooted

at α, having its edges labelled by elements of X, and satisfying the following properties:

- The vertices are the elements of the G–orbit of α. (In particular, if G is transitive, this is the whole of Ω.)
- For each edge i, j with i closer to the root α than j, there is a generator $b \in X$ labelling the edge, such that $i^b = j$. We denote such an edge by $[i, b, j]$.

In computer implementations, a Schreier tree is efficiently stored and accessed as a 'Schreier vector' (see [4]).

2.12. Constructing a Schreier Tree. A Schreier tree with root α for X is constructed as follows. Have each element of X act on α. Unless all elements of X fix α, in which case we are done, this produces a number of distinct new vertices with edges emanating from the root. Label these edges with the appropriate elements of X. We now have the vertices at distance 1 from α. Then have each element of X act on each of these vertices at distance 1. Apart from the old vertices, this produces the vertices at distance 2. Then continue in this way until the orbit is complete. At each stage, label the edges containing new vertices and be careful not to create a cycle.

For example, in Figure 2.2 you see this construction of a Schreier tree with root 1 for $\{a = (1,2)(3,4), b = (1,3)(2,4)\}$.

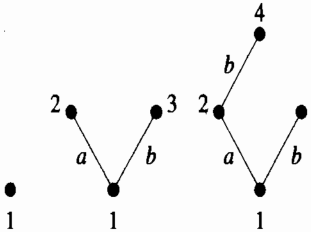


Fig. 2.2. Constructing a Schreier tree for $\{(1,2)(3,4), (1,3)(2,4)\}$.

2.13. Suppose $G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$. From a Schreier tree with root α for X we can read off a way to express a given $\omega \in \alpha^G$ as $\omega = \alpha^g$ with $g \in \langle X \rangle$. First identify the vertex ω, then follow the path down the tree until the root α is reached, while bookkeeping the generators in X labelling the edges in this path. Finally, the desired permutation is obtained by multiplying these generators in the correct order (the reverse order in which they were found); we denote this permutation by t_ω. So $\alpha^{t_\omega} = \omega$. For example, from Figure 2.2 we find $4 = 1^{ab}$ and $t_4 = ab$.

2.14. Stabilizers. Let α be an element in Ω. The problem we address next is to find a subset Y of $G = \langle X \rangle$ that generates the stabilizer G_α of α.

Elements of G that clearly stabilize α are constructed in the following way. Let T be a Schreier tree with root α for X. For $b \in X$ and $i \in \alpha^G$, the element $t_i b t_{i^b}^{-1}$ stabilizes α. Here, b can be thought of as bridging two branches of the Schreier tree, or two (possibly coinciding) vertices in the same branch (see Figure 2.3). The elements of the form $t_i b t_{i^b}^{-1}$ are called *Schreier elements* (or *Schreier generators*). (Of course, if $[i, b, i^b]$ is an edge, this construction yields the trivial element.) The relevant statement about these elements is contained in the following version of Schreier's Lemma.

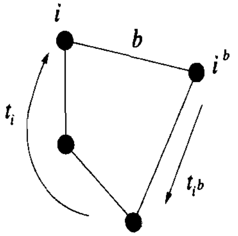


Fig. 2.3. Bridging two branches

Schreier's Lemma 2.15. *Let T be a Schreier tree with root α for X, and let $G = \langle X \rangle$. Then the stabilizer G_α of α is generated by the set of Schreier elements*

$$\{t_i b t_{i^b}^{-1} \mid i \in \alpha^G,\ b \in X\}.$$

Proof. We noted above that this set of Schreier generators is contained in the stabilizer, so we concentrate on the other inclusion.

Let $g \in G_\alpha$. Then $g = b_1 \cdots b_r$ is a product of elements $b_i \in X$ (we do not need their inverses, since we assume the group to be finite). Suppose $r > 0$. Let j be the maximal index such that $\alpha, \alpha^{b_1}, \alpha^{b_1 b_2}, \ldots, \alpha^{b_1 \cdots b_j}$ is a path in the tree with labels $b_1, \ldots, b_j$, respectively. Notice that $j < r$. Let $\beta = \alpha^{b_1 \cdots b_j}$, then $t_\beta = b_1 \cdots b_j$. Now consider $(t_\beta b t_{\beta^b}^{-1})^{-1} g$, where $b = b_{j+1}$, and rewrite it as follows:

$$(t_\beta b t_{\beta^b}^{-1})^{-1} g = t_{\beta^b} b_{j+2} \cdots b_r.$$

Then we apply the same reasoning to this element. Since t_{β^b} corresponds to a path in the tree from α, this procedure will end with an element t_γ in at most $r - j$ steps. However, as all elements at the lefthand-side of the equality stabilize α, the element t_γ has to be the trivial element. This implies that g is an element in the group generated by the Schreier generators.

2.16. Computing the Stabilizer. Schreier's Lemma suggests the following algorithm to compute stabilizers. Start with 'stab-to-be' being the empty set.

For each $b \in X$, $i \in \alpha^G$, check if $[i, b, i^b]$ is an edge. If not, insert the element $t_i b t_{i^b}^{-1}$ into the set 'stab-to-be'. Finally, 'stab-to-be' is a generating set for the stabilizer of α.

2.17. Using the algorithms so far and Proposition 2.4, we compute the order of the group $G = \langle X \rangle$, acting on $\Omega = \{1, 2, \dots, n\}$ as follows. From Proposition 2.4 we infer $|G| = |G\text{-orbit of } 1| \cdot |G_1|$. If G_1 is trivial, then we are done, since with the algorithm from 2.10 we can compute the order of an orbit. If G_1 is not trivial, then we consider the G_1-orbit of 2 and the stabilizer of 2 inside G_1, etc. Eventually we will find an i such that the stabilizer in $G_{1,\dots,i}$ of the element $i + 1$ is trivial. (This will occur after at most $n - 1$ steps. Indeed, the pointwise stabilizer of $\{1, \dots, n-1\}$ is the trivial group.) The order of G is then of course given by the product

$$|G| = |G\text{-orbit of } 1| \cdot |G_1\text{-orbit of } 2| \cdots |G_{1,\dots,i}\text{-orbit of } i + 1|.$$

2.18. Timing. Both the algorithm for computing an orbit and that for finding a generating set for a stabilizer can be performed in polynomial time with respect to the input n and the size of the generating set X. Indeed, the calculation of an orbit and the construction of the corresponding Schreier tree can be implemented to take $O(n|X|)$ steps. The application of Schreier's Lemma to obtain a generating set for a stabilizer takes (at most) $O(n^3|X|)$ steps. (Here a step is an elementary operation such as determining the image of an element of Ω under some permutation. The multiplication of two permutations and the inversion of a permutation both take $O(n)$ steps.) In particular, both algorithms are polynomial in the size of the input. However, the order algorithm, as described above, has a big disadvantage: the number of generators for the various stabilizers can become enormous. In order to avoid this we may invoke, for each stabilizer, the following algorithm 2.19, due to Sims, which guarantees at most $\binom{n}{2}$ generators for G.

Since this algorithm is polynomial in n and the size of the generating set X, also the order computation for a permutation group can be done in polynomial time. (See also 2.24 for a more efficient algorithm to compute the order of a permutation group.)

2.19. Decreasing the Number of Generators. The algorithm takes as input a generating subset X of a subgroup G of S_n. (Here we denote by G^i the pointwise stabilizer in G of $\{1, \dots, i\}$; $G^0 = G$.)

- Set $i = 1$.
  While $g, h \in X \cap G^{i-1}$ with $i^g = i^h \neq i$, replace X by $(X \setminus \{h\}) \cup \{gh^{-1}\}$ and remove any duplicates and trivial elements from X. After this step all elements in $X \cap G^{i-1}$ but not in G^i will act differently on i.
- If $i = n$, then output X and stop, otherwise increase i by 1 and go back to the previous step.

Clearly, G is still generated by the output X. However, the number of elements in X does not exceed

$$\sum_{i=1}^{n-1}(n-i) = \binom{n}{2}.$$

2.20. The algorithms described so far also provide a way to test membership in a permutation group $G \leq \mathrm{S}_n$ in polynomial time in the input parameters n and the size of the generating set. If $g \in \mathrm{S}_n$, then just compare the orders of G and $\langle G, g\rangle$ to decide whether or not $g \in G$. Although polynomial, this is not a very efficient algorithm to check membership. A better way to check membership will be discussed in 2.22.

2.3 Computing Bases and Strong Generating Sets

2.21. Bases, Stabilizer Chains and Strong Generating Sets. It is clear that, to compute the order of G, any finite sequence B of distinct points of Ω for which the (pointwise) stabilizer is trivial, suffices as input for the orbit and stabilizer computations in the algorithm from 2.17. Such a sequence is called a *base* for G. A *stabilizer chain* of G with respect to a base $B = [b_1, \ldots, b_k]$ is the chain of the subgroups $G \geq G_{b_1} \geq \cdots \geq G_{b_1,\ldots,b_{k-1}} \geq G_{b_1,\ldots,b_k} = \{1\}$, where $G_{b_1,\ldots,b_l}$ denotes the pointwise stabilizer of $b_1, \ldots, b_l$.

The combination of algorithms to compute orbits of the G—action (2.10) and stabilizers in G (2.16), which we described above, computes the order of G, but, with a little extra bookkeeping, also produces a base $[1, \ldots, i+1]$ and generators for the subgroups in the stabilizer chain for this base.

Given a base $B = [b_1, \ldots, b_k]$ and stabilizer chain $G \geq G_{b_1} \geq \cdots \geq G_{b_1,\ldots,b_{k-1}} \geq G_{b_1,\ldots,b_k} = \{1\}$ with respect to B, a generating set X for G with the property that $G_{b_1,\ldots,b_i}$ is generated by its intersection $G_{b_1,\ldots,b_i} \cap X$ with X ($i = 1, \ldots, k$), is called a *strong generating set* for G (with respect to B). Thus the above algorithm provides us with a base and, when we join all the generators found in the intermediate stages, with a strong generating set for G. Using the base B instead of $[1, \ldots, n-1]$, the algorithm in 2.19 transforms the strong generating set into one with respect to B of size at most $\binom{n}{2}$.

We notice that the construction of a base and strong generating set can be done in polynomial time in n and $|X|$.

An improved version of this algorithm is the Schreier-Sims algorithm described in 2.24. Within this algorithm we make use of the following:

2.22. Elements as Words in the Generators. Let G be a permutation group acting on $\{1, \ldots, n\}$. Suppose $B = [b_1, \ldots, b_k]$ is a base for G and $G = G^0 \geq G^1 = G_{b_1} \geq \cdots \geq G^k = G_{b_1,\ldots,b_k} = \{1\}$ is the corresponding stabilizer chain. Let X be a strong generating set for G with respect to B.

For each pair G^i and G^{i+1} we identify the G^i-orbit of b_{i+1} with the set of right cosets of G^{i+1} in G^i (see 2.5). Then we describe this action of G^i on the cosets of G^{i+1} by a Schreier tree T_{i+1}, with root b_{i+1}, for the strong generators in $X \cap G^i$. Together these Schreier trees $T_1, \ldots, T_k$ completely encode the action of G on the whole set $\{1, \ldots, n\}$.

Suppose g is an arbitrary element of $\mathrm{Sym}(\Omega)$. We now describe a procedure, called '*sifting*', which either writes g as a word in the elements of the strong generating set X for G or shows that $g \notin G$. First suppose that g fixes each base point $b_1, \ldots, b_k$. If $g = 1$ then $g \in G$ is the empty word in the strong generators, and if $g \neq 1$ then $g \notin G$. Now we may suppose that g fixes each of $b_1, \ldots, b_i$ for some $i < k$, and moves b_{i+1}. If $b_{i+1}^g \notin b_{i+1}^{G^i}$, then we conclude that $g \notin G$. Otherwise, by using the Schreier tree T_{i+1} we find elements $s_1, \ldots, s_r$ of $X \cap G^i$ such that $b_{i+1}^g = b_{i+1}^{s_1 \cdots s_r}$. Then $h := g(s_1 \cdots s_r)^{-1}$ fixes each of $b_1, \ldots, b_{i+1}$. We may now (recursively) apply the sifting procedure to h to either determine that h, and hence g, is not in G, or to find a word v in the elements of $X \cap G^{i+1}$ such that $v = h$. In the latter case, $g = vs_1 \cdots s_r$ is a word in the strong generators from X.

Exercise 2.23. Suppose both $G = \langle g_1, \ldots, g_s \rangle$ and $H = \langle h_1, \ldots, h_t \rangle$ are subgroups of S_n, given by their generators. Describe an algorithm that tests whether H is normalized by G.

Also give an algorithm to test whether $H \leq G$.

2.24. The Schreier-Sims Algorithm. C. Sims [26] devised an algorithm, now called the Schreier-Sims algorithm, to construct a base and associated strong generating set for the permutation group $G = \langle X \rangle$. This algorithm is a variant of the above, but Sims avoids the inclusion of many redundant Schreier generators in a strong generating set.

The Schreier-Sims algorithm takes as input a finite sequence B of distinct elements from Ω and a generating set S for G, such that no element of S fixes every element of B. Such a pair B, S is called a *partial base* B and *partial strong generating set* S with respect to B. (It is very easy to compute such a pair B, S given an arbitrary generating set for G.) The algorithm then attempts to verify that B is a base for G, and S a strong generating set with respect to B. If this is not the case, then the algorithm adds points to B and group elements to S, as necessary, so that on termination B is a base for G, and S is a strong generating set with respect to B.

Here is an outline of a simple version of the Schreier-Sims algorithm:

1. If $S = \{\}$, then return B, S.
2. At this point we have a nonempty partial base $B = [b_1, \ldots, b_k]$, say, and partial strong generating set S, such that $G = \langle S \rangle$, and no element of S fixes each element of B. Set $C := [b_2, \ldots, b_k]$, $T := S \cap G_{b_1}$, and apply this algorithm (recursively) with input C, T, so that they are modified to be a base $C = [b_2, \ldots, b_k, \ldots, b_l]$, say, and associated strong generating set T, for $H = \langle T \rangle$.

3. Set $B := [b_1, b_2, \ldots, b_l]$ and $S := S \cup T$. Now we can do membership testing in $H \leq G_{b_1}$ (using C and T). Using the 'sifting' algorithm from 2.22, we test each Schreier generator s for G_{b_1} to see if $s \in H$. If all such Schreier generators are in H then we are done, and return B, S.
4. Otherwise we have a Schreier generator $s \in G_{b_1}$, but $s \notin H$. We set $S := S \cup \{s\}$. If s fixes all points of B, we append to B a point of Ω which is moved by s. We now go to Step 2.

In practice, a good implementation of the Schreier-Sims algorithm can be used to compute bases and strong generating sets for permutation groups of degree up to about 10,000. There are many variations and improvements to the basic Schreier-Sims algorithm, which can extend this range (see [14, 8]).

2.25. The record of a stabilizer chain, strong generating set and corresponding Schreier trees also provides a way to systematically run through the elements of the group. In particular, we can use it to perform backtrack searches. Suppose this record for a group G is given with respect to a base $B = [b_1, \ldots, b_k]$. As before, we denote by G^i the (pointwise) stabilizer $G_{b_1,\ldots,b_i}$ of the first i elements of B; for $i = 0$, G^0 denotes G. For $1 \leq i \leq k$, let R_i be a set of right coset representatives for G^i in G^{i-1}. (The set R_i can be computed from the Schreier tree corresponding to the orbit $b_i^{G^{i-1}}$.) Then every element $g \in G$ can be expressed in a unique way as

$$g_k g_{k-1} \cdots g_1,$$

such that $g_i \in R_i$ for $i = 1, \ldots, k$ (recall 2.22). We can thus enumerate the elements of G by running through all products of this form. If we most frequently vary g_k through R_k, then g_{k-1} through R_{k-1} and so on, and if the first element in the enumeration of $g_i \in R_i$ is always taken to be the identity, we obtain an enumeration in which we first enumerate the elements of G^k, then those of G^{k-1} that are not in G^k, and so on.

Since the pointwise stabilizer of B in G is the trivial group, each element g of G is also uniquely determined by the base image B^g. Thus, instead of enumerating all elements of G as above, we could also enumerate all base images.

2.4 Generators for Some Subgroups

We sketch how Schreier's Lemma can be used to compute (generators for) various special subgroups.

2.26. An Algorithm for Orbits on Cosets. Let X be a generating set for the subgroup G of S_n. The algorithm in 2.10 presents a way to compute the orbits of G on $\{1, \ldots, n\}$. This algorithm is easily generalized to the following algorithm that constructs the permutation representation of G on the right cosets of a subgroup H of G, provided we are able to test membership in H.

This membership test makes it possible to check whether two elements g and g' represent the same coset, since $Hg = Hg'$ if and only if $gg'^{-1} \in H$.

Suppose H is such a subgroup with a membership test. Then the following algorithm, which resembles the one in 2.10, computes the G-orbit of the element H of $H\backslash G$ by means of coset representatives and a Schreier tree for the action on the right cosets.

1. Start with the set $\mathcal{O} = \{e\}$, where e is the identity element of G, representing the coset H. This is the initialization of 'orbit-to-be', the set that is to become a full set of coset representatives. Also, initialize a Schreier tree T with root e (and no other vertices at present).
2. Have each element of X act on e by multiplication on the right. If this does not produce any representatives of new cosets, we are done. Else, put the elements representing distinct new cosets in a set 'new', and update the Schreier tree T with these new elements and appropriate edge-labels.
3. Enlarge the 'orbit-to-be' $\mathcal{O}$ by taking the union of $\mathcal{O}$ and the set 'new'.
4. Have each element of X act on each element of 'new' (rather than of 'orbit-to-be' as this saves work) by multiplication on the right. If this does not produce any cosets not already represented in $\mathcal{O}$, then we are done. Else set 'new' to contain precisely the elements found at this stage representing distinct cosets that are not already represented by elements in $\mathcal{O}$, and update the Schreier tree T accordingly.
5. Go back to 3.

Together with Schreier's Lemma, this algorithm can be used to compute a set of generators for H algorithmically (cf. 2.16): just compute (generators for) the stabilizer of the element e representing the coset $He = H$.

2.27. Subgroups with Membership Test. The above algorithm depends on the presence of a membership test for the subgroup H of G. As examples of groups for which we have a membership test, one can think of:

- The centralizer of an element or of a subgroup (given by a set of generators) of G.
- The center of a group G generated by a set of permutations from S_n.
- The (setwise) stabilizer in G of some subset of $\{1, \ldots, n\}$.
- The normalizer of some subgroup K for which we have a membership test.
- The intersection of two subgroups G and H of S_n, provided we have a generating set for G and a membership test for H. Apply the algorithm to construct the permutation representation of G acting on the right cosets of $G \cap H$ in G. Schreier's Lemma then yields a generating set for $G \cap H$, and the 'sifting'-algorithm provides us with a membership test for $G \cap H$.

Remark 2.28. Suppose the membership test for H can be performed in time polynomial in n. Then the above algorithm runs in time polynomial in n, the size of the set of generators for G, and the size of $\mathcal{O}$, which is the index of H in G. However, this index is often huge, and other methods, such as backtrack search, to compute a generating set for the subgroup H are often much preferable (see [4, 15, 16]).

2.29. Here is a further list of special subgroups. Some of them are used in deciding nilpotency or solvability of a permutation group given by a set of generating permutations.

2.30. Normal Subgroups. Let $G = \langle X \rangle$ and $H = \langle Y \rangle$ be two subgroups of S_n. Since, with the help of the 'sifting'-algorithm, we are able to test membership of elements $y \in Y$ of G, we can check whether H is a subgroup of G. Moreover, we can even check whether H is a normal subgroup in G by checking the membership of $x^{-1}yx$ in H for all $x \in X$ and $y \in Y$.

2.31. Normal Closure. For a subgroup $H = \langle Y \rangle$ of $G = \langle X \rangle$ the *normal closure* of H in G is the smallest normal subgroup of G containing H. The normal closure of H in G is generated by $\{g^{-1}yg \mid g \in G, y \in Y\}$.

The following is an algorithmic approach to finding a generating set S for the normal closure of H in G.

- Start with setting $S := Y$.
- As long as there is an element $s \in S$ and $x \in X$ with $x^{-1}sx \notin \langle S \rangle$ replace S by $S \cup \{x^{-1}sx\}$.
- Return S.

The final set S thus obtained generates a normal subgroup of G and hence is the normal closure of H in G.

2.32. The Commutator Subgroup. The *commutator subgroup* $G' = [G,G]$ of $G = \langle X \rangle$ is the normal subgroup generated by all commutators $[g,h] = g^{-1}h^{-1}gh$, with $g,h \in G$. In particular, the normal closure of the subgroup $\langle [x,x'] \mid x,x' \in X \rangle$ is certainly contained in G'. Since $[xy,z] = y^{-1}[x,z]y[y,z]$ for all $x,y,z \in G$, we see that every commutator of $G = \langle X \rangle$ is contained in the normal closure of $\langle [x,x'] \mid x,x' \in X \rangle$. Hence this normal closure equals the commutator subgroup of G. Therefore we can compute the commutator subgroup of G with the help of the above algorithm.

If $H = \langle Y \rangle$ is a subgroup of $G = \langle X \rangle$, then similarly we can compute the subgroup $[G,H] = \langle [g,h] \mid g \in G, h \in H \rangle$ as the normal closure of $\langle [x,y] \mid x \in X, y \in Y \rangle$ in G.

2.33. Solvable and Nilpotent Groups. The commutator subgroup of G is also called the (first) derived subgroup of G. Inductively one defines the n-th derived group $G^{(n)}$ of G as the commutator subgroup of the $(n-1)$-th derived subgroup. The *commutator or derived series* of G is the chain of subgroups

$$G \geq G^{(1)} \geq G^{(2)} \geq \dots$$

(Continue this series until it is stable.) The group G is said to be *solvable* if this series terminates with the trivial group $\{1\}$.

By Lagrange's Theorem 2.6 a commutator series can have length at most ${}^2\mathrm{Log}(|G|)$. Hence, as we can compute commutator subgroups, we can compute the commutator series of a permutation group G given by a set of generators and test it for solvability. Moreover, since the computing of normal closures and commutator groups can be done in polynomial time and the length of a commutator series is at most ${}^2\mathrm{Log}(n!)$, the commutator series can be calculated in polynomial time.

The *lower central series* of a group G is the chain of subgroups

$$L_0 \geq L_1 \geq L_2 \geq \dots$$

where $L_0 = G$ and $L_{i+1} = [G, L_i]$. The group G is called *nilpotent* if the lower central series of G terminates at the trivial group. Similar to the computation of the derived series of G we can compute the lower central series of G and test G for nilpotency in polynomial time.

3. Coset Enumeration

3.1 Introduction

The starting point in this section is a group given by a finite presentation by generators and relators. We explain how to obtain a permutation representation of such a group on the set of cosets of a subgroup of finite index. The procedure we discuss is due to Todd and Coxeter [29] and is known as *(Todd-Coxeter) coset enumeration*. Coset enumeration and related algorithms for finitely presented groups are available in the group theory system GAP and in the algebra system MAGMA.

3.2 Todd-Coxeter Coset Enumeration

Let G be a group given by a finite set $X = \{g_1, \dots, g_n\}$ of generators, subject to a finite set R of *relators* in these generators. (Here each element $r \in R$ is a word in the elements of $X \cup X^{-1}$, and saying that r is a relator is the same thing as saying that $r = 1$ is a relation.) Thus, G is the quotient of the free group on $\{g_1, \dots, g_n\}$ by the normal closure of the subgroup generated by the elements in R. This will be denoted by

$$G = \langle X \mid R \rangle.$$

Suppose H is a subgroup of G generated by $Y = \{h_1, \dots, h_t\}$, and that all the elements of Y are given as words in the elements of $X \cup X^{-1}$. We will discuss a method, first described by Todd and Coxeter [29], which, if it terminates, provides us with the permutation representation of G on the right cosets of the subgroup H. The Todd-Coxeter coset enumeration method is a trial and error process that tries to enumerate all the different cosets of H in G. These cosets will be denoted by positive integers; the integer 1 represents H. The notation n^g is shorthand for the (label of the) image under g of the coset with label n.

Todd-Coxeter enumeration relies on the following three observations:

- **TC–1**: $1^h = 1$ for all $h \in Y$;
- **TC–2**: $j^r = j$ for all cosets j and $r \in R$;
- **TC–3**: $i^g = j \Leftrightarrow i = j^{g^{-1}}$ for all cosets i, j and all $g \in X$.

These observations will be used in setting up three kinds of tables for the action (by multiplication on the right) of the elements of X and X^{-1} on the set $H\backslash G$ of right cosets of H in G. The entries of the tables will be filled with the various (integers representing the) cosets of H.

In explaining the three kinds of tables, we will illustrate the process for the group

$$G = \langle x, y \mid x^2, y^2, (xy)^3 \rangle,$$

isomorphic to S_3, and its subgroup H generated by x.

3.1. Three Types of Tables. First, for every generator $h = g_{j_1} \cdots g_{j_l} \in Y$ of H, where $g_{j_i} \in X \cup X^{-1}$, we construct a so-called *subgroup table.* This table consists of only one row of length $l+1$ and starts and ends with the entry 1 (this corresponds to **TC–1**: multiplying the coset H with h gives back H again, see 3.2 below). The last l entries of this row are indexed by g_{j_1} up to g_{j_l}. The table is set up in order to describe the action of $g_{j_1}, g_{j_1}g_{j_2}, \ldots, h = g_{j_1} \cdots g_{j_l}$ on the coset H. In our example there is only one subgroup table, reflecting the equality $Hx = H$:

subgroup	x
1	1

Second, for each relator $r = g_{i_1} \cdots g_{i_k} \in R$, with $g_{i_j} \in X \cup X^{-1}$, we construct a *relator table* with $k+1$ columns, the last k of which are indexed by $g_{i_1}, \ldots, g_{i_k}$. The number of rows is determined during the process. Again, each row starts and ends with the same integer, reflecting **TC–2**; each row is filled with the images of the coset corresponding to this integer under g_{i_1}, $g_{i_1}g_{i_2}, \ldots, r = g_{i_1} \cdots g_{i_k}$, respectively, as will be explained in more detail in 3.2.

For our specific group, there are three relator tables; using the subgroup table, the first row of each of these is filled as follows (a new coset is introduced if no information is available to fill in a spot):

relator	x	x
1	1	1

relator	y	y
1	2	1

relator	x	y	x	y	x	y
1	1	2	3	4	5	1

Third, for bookkeeping of the desired permutation representation, we construct a *coset table* consisting of $|X|+1$ columns, where the last $|X|$ columns are indexed by the elements of X. The unlabelled column contains the integers already assigned to a coset in $H\backslash G$ during the process. In the rows we store the various images of a coset under multiplication by the elements of X. In particular, the g-th entry of the row starting with k contains the integer representing the coset k^g, if k^g has been defined, and otherwise some symbol (or blank) is used to indicate that k^g has not yet been defined. (For efficiency, in most machine implementations of coset enumeration the coset table also contains a column for each element of X^{-1} that is not obviously in X. Also in machine implementations, the subgroup and relator tables are not stored explicitly, but information in them is recreated as necessary. See [22, 5].)

This leads, at this stage, to the following coset table for G:

coset	x	y
1	1	2
2	3	1
3		4
4	5	
5		1

It is clear that in the case of the group G there cannot be five cosets. In fact, as we will see below, the cosets labelled 2 and 5 coincide as well as the cosets labelled 3 and 4. So we turn to a more detailed description of how to fill the various tables and how to 'scan' for such 'coincidences'.

3.2. Filling in the Tables. The basic idea is to fill the subgroup and relator tables so that the following holds: if two neighbouring spots in a row are filled from left to right with (integers representing) the cosets H' and H'', and if H'' is in the column indexed by g, then $H'g = H''$ (it is sometimes convenient to read this as $H' = H''g^{-1}$, **TC–3**).

	g
H'	$H'' = H'g$

Once we discover that $H'g = H''$ from a subgroup or relator table, we may need to update the coset table. If $g \in X$ and there is no entry in the coset table recording the image of H' under g, we record there that $H'g = H''$. Similarly, if $g^{-1} \in X$ and there is no entry in the coset table recording the image of H'' under g^{-1}, we record there that $H''g^{-1} = H'$. (If the coset table already contains information which implies there are two different integers representing H' or H'', we have obtained a coincidence, the processing of which is described later.)

In each relator table, the first entry of the first row is filled with a 1. Since for each relator r we have that $j^r = j$, each row starts and ends with the same integer.

In a subgroup or relator table, an empty entry to the right of an entry m, where the empty entry is in the column indexed by $g \in X \cup X^{-1}$, is filled with m^g if this coset is already known by the information in the coset table; otherwise it can be filled with the smallest positive integer s not yet used. In that case, we add a new row starting with s to the coset table as well as the information $m^g = s$ (possibly in the form $s^{g^{-1}} = m$). We also add a new row starting with s to each relator table. A similar action is taken to fill an empty spot to the left of an entry m: if g indexes the column containing m, then the open spot is filled with $m^{g^{-1}}$ if this coset already has a label and by the smallest unused positive integer otherwise. Again this information and a new row are added to the coset table and a new row is added to each relator table.

3.3. Scanning for Coincidences. Let's return to our example first. Adding the rows for the cosets labelled 2 to 5 and filling some of the obvious spots, we obtain

relator	x	x
1	1	1
2	3	2
3		3
4	5	4
5		5

relator	y	y
1	2	1
2	1	2
3	4	3
4		4
5	1	5

relator	x	y	x	y	x	y
1	1	2	3	4	5	1
2	3	4	5	1	1	2
3						3
4						4
5						5

(In fact, all entries can be filled now, but we don't need this yet.)

From the second and third relator tables we obtain that $2^y = 1$ and $5^y = 1$. This says that the cosets labelled 2 and 5 are in fact equal. So we replace 5 by 2 in the relator tables and remove the rows starting with a 5, so the tables then collapse to tables with four rows. Another collapse is the following: from the first relator table we deduce $2^x = 3$ and $2^x = 4$, so $3 = 4$. Only three cosets remain and it is straightforward to finish the relator tables

relator	x	x
1	1	1
2	3	2
3	2	3

relator	y	y
1	2	1
2	1	2
3	3	3

relator	x	y	x	y	x	y
1	1	2	3	3	2	1
2	3	3	2	1	1	2
3	2	1	1	2	3	3

and the coset table

coset	x	y
1	1	2
2	3	1
3	2	3

In particular, we have found a permutation representation of G into S_3, where x is mapped to $(2,3)$ and y to $(1,2)$. The index of H in G equals 3. As H is a group of order at most 2, the group G is of order at most 6. So the above permutation representation establishes an isomorphism between G and S_3.

When filling in the entries of the various tables, we may implicitly be writing down relations between some of the cosets, just like in the example.

Such an event where two distinct integers i and j are found to represent the same coset is called a *coincidence* (or *collapse*) of i and j. This happens when the information in our tables tells us that $i^g = j^g$ for some $g \in X \cup X^{-1}$, but $i \neq j$.

If a coincidence is discovered then it must be processed. Here is a basic way to process a coincidence of $i < j$. First, for each $g \in X$ for which i^g is not defined in the coset table but j^g is, insert in the coset table the information that $i^g = j^g$. Next, replace each entry j in all the tables by i. Then check all filled entries of the tables for further coincidences and repeat this procedure until there are no more coincidences. Finally, in each table, for each coset i, remove all rows starting with i except the first such. Notice that this procedure is finite as the occurrence of each coincidence reduces the number of integers used in the tables by one.

Efficient and correct handling of coincidences is the most delicate part of a practical computer implementation of coset enumeration (see [22, 5] for useful details).

Todd-Coxeter coset enumeration procedures consist of filling in the tables and scanning for and processing coincidences until all tables are completely filled and no more coincidences can be deduced from the tables. Note that at various points in the procedure, there may be more than one way to continue. There is an enormous amount of flexibility in the coset enumeration process, and many different approaches have been suggested and experimented with (see [5, 22]). Depending on the presentation and the method used, there can be huge variations in the time and store taken by a coset enumeration. At present, the most advanced methods are due to George Havas, and these methods are available in the MAGMA [3] system.

If a Todd-Coxeter coset enumeration procedure terminates, then the final coset table gives a set of permutations that satisfy the three conditions **TC–1** to **TC–3**.

Of course, termination of the procedure is possible only if the index of H in G is finite. But the converse is also true, provided each row of each table is filled (or deleted) after a finite number of steps and provided the tables are regularly scanned for coincidences. We state this in the theorem below, first proven by Mendelsohn [20]. (Although termination of Todd-Coxeter coset enumeration can be guaranteed when the index of H in G is finite, there can be no general bound on the time or store required for such an enumeration. This follows from the fact that determining whether or not a finitely presented group is trivial is algorithmically undecidable.)

For the purposes of this theorem we assume that we also explicitly maintain relator tables for the trivial relators gg^{-1} and $g^{-1}g$, for all $g \in X$. This is to ensure that i^g and $i^{g^{-1}}$ are eventually defined for each coset i and each $g \in X$.

Theorem 3.4. ***Suppose the index of H in G is finite. Any Todd-Coxeter coset enumeration procedure in which it is taken care of that a) each row of***

each table is completely filled (or deleted) after a finite number of steps and that b) there are only finitely many steps between two scannings of the tables for coincidences, will terminate.

Proof. Consider the first row of any table. After a finite number of steps all the entries of this row are filled. The first entry, 1, is stable, and the other entries can only change to smaller integers. This can only happen a finite number of times, so the row remains stable after a finite number of steps.

Assume that after a finite number of steps all the first $k-1$ rows of the tables are filled and stable, and that ρ is a k-th row of one of the relator tables. Let a be the first entry of ρ. Then there exists an element $b < a$ occurring in one of the stable rows and a $g \in X \cup X^{-1}$ with $b^g = a$. (Indeed, a was first defined as $(b')^g$ for some $b' < a$, and collapses can only replace this b' by some integer $b \leq b'$.) So a occurs somewhere among the first $k-1$ rows and is therefore stable. Then, as we argued for the first row, ρ will also be stable after a finite number of steps.

So if the procedure does not terminate after a finite number of steps, the number of stable rows will increase beyond any bound. In particular, the procedure would provide us with a transitive permutation action of G on a countably infinite set (just read the coset table) in which H is contained in the stabilizer. This contradicts the fact that H has finite index in G.

Exercise 3.5. Perform coset enumeration on the above example $G \simeq S_3$, but try to choose a strategy that does not force you to define more than 3 cosets.

Example 3.6. Let

$$G = \langle a, b, c \mid a^3, b^2, c^2, (ab)^3, (ac)^2, (bc)^3 \rangle.$$

We will perform coset enumeration with respect to the subgroup generated by a and b. We begin with the following tables:

subgroup	a
1	1

subgroup	b
1	1

relator	a	a	a
1	1	1	1

relator	c	c
1	2	1
2	1	2

relator	b	b
1	1	1
2	3	2
3	2	3

Now we add the tables for $(ac)^2$ and $(bc)^3$ and fill them as far as possible:

relator	a	c	a	c
1	1	2		1
2	2	1	1	2
3				3

relator	b	c	b	c	b	c
1	1	2	3	3	2	1
2	3	3	2	1	1	2
3	2	1	1	2	3	3

Implicitly we have obtained that $2^a = 2$ and $3^c = 3$. After adding two cosets $3^a = 4$ and $4^a = 5$ we get:

relator	a	a	a
1	1	1	1
2	2	2	2
3	4	5	3
4	5	3	4
5	3	4	5

relator	c	c
1	2	1
2	1	2
3	3	3
4		4
5		5

relator	a	c	a	c
1	1	2	2	1
2	2	1	1	2
3	4		3	3
4	5			4
5	3	3	4	5

From these we deduce that $4^c = 5$. This leads to:

relator	a	a	a
1	1	1	1
2	2	2	2
3	4	5	3
4	5	3	4
5	3	4	5

relator	c	c
1	2	1
2	1	2
3	3	3
4	5	4
5	4	5

relator	a	c	a	c
1	1	2	2	1
2	2	1	1	2
3	4	5	3	3
4	5	4	5	4
5	3	3	4	5

relator	b	b
1	1	1
2	3	2
3	2	3
4		4
5		5

relator	b	c	b	c	b	c
1	1	2	3	3	2	1
2	3	3	2	1	1	2
3	2	1	1	2	3	3
4					5	4
5					4	5

So far we have not used the relator table for $(ab)^3$ yet. This table can already be filled as follows:

relator	a	b	a	b	a	b
1	1	1	1	1	1	1
2	2	3	4		3	2
3	4	5	3	2	2	3
4	5					4
5	3	2	2	3	4	5

and we obtain $4^b = 5$. It is easy now to fill all tables consistently with the numbers 1 up to 5 and so the group $\langle a, b \rangle$ has index 5 in G. Moreover, we obtain a permutation representation of G into A_5. This representation is actually an isomorphism as follows from Exercise 3.7 below.

Exercise 3.7. Let H be the group given by the following presentation

$$H = \langle a, b \mid a^3, b^2, (ab)^3 \rangle.$$

Perform coset enumeration with respect to the subgroup $\langle a \rangle$, and show that $H \simeq A_4$.

Example 3.8. We end this section with a larger example. Since it is too elaborate to easily do the computations in this example by hand, we have performed them with the help of GAP.

Consider the following presentation:

$$G \doteq \langle a, b, c, d \mid a^3, b^2, c^2, d^2, (ab)^3, (ac)^2, a(cd)^4, (bc)^3, (bd)^2 \rangle.$$

Coset enumeration with respect to the subgroup $\langle a, b, c\rangle$ provides us with a transitive permutation representation of the group G of degree 22. The elements a, b, c and d are mapped to the following permutations α, β, γ and δ, respectively:

$$\begin{aligned} \alpha &= (4,6,8)(7,10,9)(11,12,13)(14,18,16)(15,17,20)(19,22,21), \\ \beta &= (3,4)(5,7)(6,8)(9,10)(12,15)(13,17)(16,19)(18,21), \\ \gamma &= (2,3)(6,8)(7,11)(9,12)(10,13)(14,18)(15,17)(19,22), \end{aligned}$$

and

$$\delta = (1,2)(3,5)(4,7)(6,9)(8,10)(11,14)(12,16)(13,18)(15,19)(17,21)(20,22).$$

The group $\langle \alpha, \beta, \gamma\rangle$, which by the previous example is isomorphic to A_5, stabilizes the point 1 and has three orbits on the remaining points; one of length 5, one of length 10 and one of length 6. The group G has order $22 \cdot |A_5| = 1320$. Let

$$P = \{1, 5, 7, 9, 10, 11, 12, 13, 15, 17, 20\} = \{1\} \cup 5^{\langle \alpha,\beta,\gamma\rangle}$$

and

$$B = \{1, \ldots, 22\} \setminus P = 2^{\langle \alpha,\beta,\gamma\rangle} \cup 14^{\langle \alpha,\beta,\gamma\rangle}.$$

Then G stabilizes the partition $\{P, B\}$ of $\{1, \ldots, 22\}$; the permutation δ interchanges P and B. The stabilizer H of one part, say P, has index 2 in G and is 2-transitive on the 11 points in P. It is a fact that H is a simple group isomorphic to $PSL_2(11)$. We will encounter this group again in Project 6 of this book.

Notes

Permutation Groups

The basic notions on symmetric groups can be found in almost any algebra textbook. The basics of permutation representations, like orbit and stabilizer, are described in [23].

Schreier's Lemma goes back to [25], where Schreier gives a proof of the celebrated Nielsen-Schreier Theorem that subgroups of free groups are free, see also [10].

The concepts of base and strong generating set were introduced by Sims [26, 27]. They form the key ingredients for most permutation group algorithms. The algorithm 2.21 to find a base and strong generating set is also due to Sims [26].

For several aspects that we did not discuss we refer to the literature: [2] and [19] for complexity, [1] and its references for probabilistic approaches, and [4, 15, 16] for backtrack search algorithms.

An in-depth discussion of polynomial-time permutation group algorithms is given in [19]. For example, the centre of a permutation group can be found in polynomial time. One deep result in polynomial-time group theory is that a Sylow p-subgroup of a permutation group can be found in polynomial time. This was proved by Kantor (see [11] and Kantor and Taylor [12]) using the classification of finite simple groups. Although Kantor's algorithm does not seem to be very practical, significant progress by Morje [21] lays the theoretical groundwork to what should be a practical polynomial-time algorithm for Sylow subgroups.

Coset Enumeration

Groups given by generators and relations play a central role in group theory, but they are also used in different branches of mathematics, such as knot theory, see for example [7]. A good reference for the theory of group presentations is [10].

Coset enumeration is due to Todd and Coxeter, see [29]. A good impression of the usefulness of Todd-Coxeter coset enumeration, and more detailed discussion of this process, its computer implementation and its variants, can be obtained from [5, 13, 22, 10, 28]. Mendelsohn's result on the termination of Todd-Coxeter coset enumeration was first described in [20].

We have described a very basic hand approach to coset enumeration. Most coset enumeration is now done by quite sophisticated computer programs, and it is not uncommon to enumerate 10^6 or more cosets by machine computation.

Double coset enumeration is described by Linton in [17], and he has also devised an algorithm [18] closely related to coset enumeration for constructing matrix (rather than permutation) representations of finitely presented groups.

Computer Algebra Systems and Algorithms

It is quite straightforward to write your own programs, say in Maple, for the algorithms discussed in the first section (this has already been done for most of these algorithms in the Maple package *group*). In GAP and MAGMA, everything we discussed and much more is available, efficient and a pleasure to work with.

For an up-to-date overview of the forefront of research in computational group theory we refer the reader to [9], which includes papers on new approaches to computing with groups generated by matrices.

References

1. L. Babai (1996): *Randomization in group algorithms: Conceptual questions*, pp. 1–17 in Groups and Computation II (L. Finkelstein and W.M. Kantor, eds), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **28**, American Math. Soc.

2. W. Bosma and J. Cannon (1992): *Structural computation in finite permutation groups*, CWI Quarterly **5** (2) 127–160.
3. W. Bosma, J. Cannon, and G. Matthews (1994): *Programming with algebraic structures: the design of the Magma language*, pp. 52–57 in Proceedings of ISSAC '94, Assoc. Comp. Mach.
4. G. Butler (1991): *Fundamental algorithms for permutation groups*, Lecture Notes in Computer Science **559**, Springer-Verlag, Berlin Heidelberg New York.
5. J. J. Cannon, L. A. Dimino, G. Havas, and J. M. Watson (1973): *Implementation and analysis of the Todd-Coxeter algorithm*, Math. Comp. **27**, 463–490.
6. B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt (1992): *First Leaves: A Tutorial Introduction to Maple V*, Springer-Verlag, Berlin Heidelberg New York.
7. R. H. Crowell and R. H. Fox (1977): *Introduction to Knot Theory*, Graduate Texts in Mathematics **57**, Springer-Verlag, Berlin Heidelberg New York.
8. L. Finkelstein and W. M. Kantor (editors) (1993): *Groups and Computation*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **11**, American Math. Soc.
9. L. Finkelstein and W. M. Kantor (editors) (1996): *Groups and Computation II*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **28**, American Math. Soc.
10. D.L. Johnson (1990): *Presentations of Groups*, Cambridge University Press, Cambridge.
11. W. M. Kantor (1985): *Sylow's theorem in polynomial time*, J. Comput. System Sci. **30**, 359–394.
12. W. M. Kantor and D. E. Taylor (1988): *Polynomial-time versions of Sylow's theorem*, J. Algorithms **9**, 1–17.
13. J. Leech (1984): *Coset enumeration*, pp. 3–18 in Computational Group Theory (M.D. Atkinson, ed.), Academic Press, London.
14. J.S. Leon (1980): *On an algorithm for finding a base and strong generating set for a group given by generating permutations*, Math. Comp. **35**, 941–974.
15. J.S. Leon (1991): *Permutation group algorithms based on partitions, I: theory and algorithms*, J. Symb. Comput. **12**, 533–583.
16. J.S. Leon (1996): *Partitions, refinements, and permutation group computation*, pp. 123–158 in Groups and Computation II (L. Finkelstein and W.M. Kantor, eds), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **28**, American Math. Soc.
17. S.A. Linton (1991): *Double coset enumeration*, J. Symb. Comput. **12**, 415–426.
18. S.A. Linton (1991): *Constructing matrix representations of finitely presented groups*, J. Symb. Comput. **12**, 427–438.
19. E.M. Luks (1993): *Permutation groups and polynomial-time computation*, pp. 139–175 in Groups and Computation, (L. Finkelstein and W.M. Kantor, eds), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **11**, American Math. Soc.
20. N.S. Mendelsohn (1965): *An algorithmic solution for a word problem in group theory*, Canad. J. Math. **16**, 509–516. Correction, Canad. J. Math. **17**, 505.
21. P. Morje (1996): *On nearly linear time algorithms for Sylow subgroups of small-base permutation groups*, pp. 257–272 in Groups and Computation II (L. Finkelstein and W.M. Kantor, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **28**, American Math. Soc.
22. J. Neubüser (1982): *An elementary introduction to coset table methods in computational group theory*, pp. 1–45 in Groups – St. Andrews 1981 (C.M. Campbell and E.F. Robertson, eds), LMS Lecture Notes **71**, Cambridge University Press, Cambridge.

23. P. M. Neumann, G. A. Stoy, and E. C. Thompson (1994): *Groups and Geometry*, Oxford University Press, Oxford.
24. M. Schönert, et al. (1994): GAP – *Groups, Algorithms and Programming, version 3, release 4*, Lehrstuhl D für Mathematik, RWTH Aachen.
25. O. Schreier (1927): *Die Untergruppen der freien Gruppen*, Abh. Math. Sem. Univ. Hamburg **5**, 161–183.
26. C.C. Sims (1971): *Computation with permutation groups*, pp. 23–28 in Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation (S.R. Petrick, ed.), Assoc. Comp. Mach.
27. C. C. Sims (1971): *Determining the conjugacy classes of a permutation group*, pp. 191–195 in SIAM-AMS Proceedings **4**, American Math. Soc.
28. C. C. Sims (1994): *Computation with Finitely Presented Groups*, Cambridge University Press, Cambridge.
29. J. A. Todd and H. S. M. Coxeter (1936): *A practical method for enumerating cosets of a finite abstract group*, Proc. Edinburgh Math. Soc. **5**, 26–34.

Chapter 9. Symbolic Analysis of Differential Equations

Marius van der Put

1. Introduction

The purpose of this chapter is to give an idea of the methods for solving linear differential equations with 'computer algebra'. Sections 2, 3, and 4 are elementary and use almost no differential algebra.

The later sections are less elementary since differential Galois theory plays an essential role there. Some standard differential algebra is presented in Section 5 as well as some highlights of differential Galois theory. There are almost no proofs given.

Kovacic's algorithm ([3]) for order two equations is explained in some detail in Section 6. It can be seen as a very concrete application of differential Galois theory. For later use the local differential Galois groups of the equation $y'' = ry$ are calculated (including complete proofs) in Section 7.

The next section studies the special case of an order two equation with only one singular point.

The simplifications of Kovacic's algorithm for order two equations with two singular points is the subject of Section 9.

2. The Equation $y' = f$ with $f \in C(x)$

Some Background and Notation

By C we denote a field of characteristic 0, i.e., the field C contains the field $\mathbb{Q}$ of rational numbers. Furthermore, $\overline{C}$ stands for an algebraic closure of C. The field of rational functions with coefficients in C is denoted by $C(x)$. This field consists of the expressions $\frac{a}{b}$ where $a, b \in C[x]$ are polynomials (and $b \neq 0$). The operation $' = \frac{d}{dx}$ on $C(x)$ is the usual differentiation of rational functions. Similarly, one considers the differentiation $' = \frac{d}{dx}$ on the field $\overline{C}(x)$. The derivatives of a function y are denoted by $y', y'', \ldots$ or by $y^{(n)}$ for $n > 0$.

Sometimes new functions, e.g., $\log(v)$ with $v \in \overline{C}(x)$, are added to the field $\overline{C}(x)$ in order to express solutions of the differential equation under consideration. We will work intuitively with those expressions and we will use the rules $\log(vw) = \log(v) + \log(w)$ and $\log(v)' = \frac{v'}{v}$. Those expressions can be interpreted in the case where C is a subfield of the field of complex numbers $\mathbb{C}$, as actual logarithms of rational functions.

The 'calculus approach' to the equation $y' = f$ with $f \in C(x)$ is as follows. The partial fraction decomposition of f is a finite sum

$$f = p + \sum_{n,\alpha} \frac{c(n,\alpha)}{(x-\alpha)^n},$$

with p a polynomial and all $\alpha, c(n,\alpha) \in \overline{C}$. Then there is a solution $y \in \overline{C}(x)$ if and only if $c(n,\alpha) = 0$ for $n = 1$ and all α. If some $c(1,\alpha)$'s are not zero then one can still write a solution in closed form, using some $\log(x-\alpha)$. This method has two disadvantages:

- This closed form solution is often very complicated.
- One has to calculate the poles of the rational function f.

Our aim is to find a symbolic formula for y which uses the minimal amount of algebraic numbers.

We need the notion of *resultant*. Let $f = f_m x^m + \cdots + f_1 x + f_0$ and $g = g_n x^n + \cdots + g_1 x + g_0$ be two polynomials in x with coefficients in some field. The resultant $R_x(f,g)$ is defined as the determinant of the $(n+m)$-matrix

$$\begin{pmatrix}
f_m & \cdot & \cdot & \cdot & f_0 & & & & \\
 & f_m & \cdot & \cdot & \cdot & f_0 & & & \\
 & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\
 & & & \cdot & \cdot & \cdot & \cdot & \cdot & \\
 & & & & f_m & \cdot & \cdot & \cdot & f_0 \\
g_n & \cdot & \cdot & \cdot & g_0 & & & & \\
 & g_n & \cdot & \cdot & \cdot & g_0 & & & \\
 & & \cdot & \cdot & \cdot & \cdot & \cdot & & \\
 & & & & g_n & \cdot & \cdot & \cdot & g_0
\end{pmatrix}$$

One can prove the following formula

$$R_x(f,g) = f_m^n g_n^m \prod_{1 \le i \le m, 1 \le j \le n} (\alpha_i - \beta_j),$$

where the α_i and the β_j are the zeros of f and g (in some field extension). Hence the resultant is zero if and only if f and g have a common root (in some field extension).

The algorithm that we will give is part of the Risch algorithm. There are contributions of Hermite, Rothstein, Trager and many others to the algorithm.

2.1 The Algorithm

Given $f \in C(x)$, the algorithm finds $y \in C(x)$ with $y' = f$ or reports nonexistence of such a y. In the latter case, the algorithm produces the simplest formula for y. This formula involves rational functions, some logarithms of rational functions and some constants, algebraic over the field C.

(1) Write $f = \frac{r}{q} + s$ with r, q, s polynomials, q monic, $\gcd(r, q) = 1$ and the degree of r is less than the degree of q. Let $q = q_1 q_2^2 q_3^3 \cdots q_k^k$ denote the *square-free decomposition*, i.e., the q_i are monic polynomials and $q_1 q_2 q_3 \cdots q_k$ has no multiple zeros. The q_i are calculated using gcd's only. With linear algebra one can find an expression

$$\frac{r}{q} = \sum_{1 \le i \le k,\ 1 \le j \le i} \frac{r_{i,j}}{q_i^j},$$

with $r_{i,j}$ polynomials. For $j > 1$ one writes $r_{i,j} = sq_i + tq_i'$. This is possible since q_i and q_i' are relatively prime. The Euclidean algorithm produces s and t. Then

$$\frac{r_{i,j}}{q_i^j} = \frac{s + (j-1)^{-1} t'}{q_i^{j-1}} + \left(\frac{t}{(1-j) q_i^{j-1}} \right)'.$$

With steps of this type we find

$$\int \frac{r}{q} dx = \sum_{1 \le i \le k} \int \frac{r_i}{q_i} dx + \text{ a rational function with rational coefficients.}$$

This reduces the general case to $\int \frac{a}{b} dx$ with a, b polynomials, b monic, the degree of a less than the degree of b and $\gcd(a, b) = \gcd(b', b) = 1$. This is the starting point of the next step.

(2) We now focus on $\int \frac{a}{b} dx$. For notational convenience we suppose that $C = \mathbb{Q}$. This is anyway the most interesting case. The roots of any polynomial with coefficients in $\mathbb{Q}$ are considered as elements of the field of complex numbers $\mathbb{C}$. We introduce a new variable y and form

$$R(y) := R_x(a - yb', b) \in \mathbb{Q}[y].$$

For a complex number γ one has $R(\gamma) = 0$ if and only if the $\gcd(a - \gamma b', b) \neq 1$. Let $\gamma_1, \ldots, \gamma_n \in \mathbb{C}$ denote the distinct zeros of $R(y)$.

Define $v_i := \gcd(a - \gamma_i b', b) \in \mathbb{Q}(\gamma_1, \ldots, \gamma_n)[x]$. In Proposition 2.2 we will show that $\frac{a}{b} = \sum_{i=1}^n \gamma_i \frac{v_i'}{v_i}$ and that $\int \frac{a}{b} dx = \sum_{i=1}^n \gamma_i \log(v_i)$ is '*the best formula*' for the primitive of the rational function $\frac{a}{b}$. This means that any other formula contains also the algebraic numbers $\gamma_1, \ldots, \gamma_n$ (see part (2) of Proposition 2.2). We will first give an example, which explains the meaning of the γ_i and the v_i.

Example 2.1. Let $\frac{a}{b}$ be $\frac{-6x^3-8x^2-24x+8}{15x^4-75x^2-60}$. The solutions $\gamma_1, \gamma_2, \gamma_3$ of $R(y) = 0$ are $1/3, -2/5, -2/3$. Furthermore γ_1 is a double zero of $R(y) = 0$. The v_i are $x^2-1, x+2, x-2$. Hence $\frac{a}{b} = \frac{1}{3}\frac{2x}{x^2-1} - \frac{2}{5}\frac{1}{x+2} - \frac{2}{3}\frac{1}{x-2}$. The first term can be expanded a little further as $\frac{1}{3}\frac{1}{x+1} + \frac{1}{3}\frac{1}{x-1}$. The complete decomposition of $\frac{a}{b}$ as partial fraction is:

$$\frac{1}{3}\frac{1}{x+1} + \frac{1}{3}\frac{1}{x-1} - \frac{2}{5}\frac{1}{x+2} - \frac{2}{3}\frac{1}{x-2}.$$

One observes that the γ_i are the 'residues' of $\frac{a}{b}\ dx$ at the various poles of $\frac{a}{b}\ dx$. Moreover v_i is the product of the terms $x - \alpha$ occurring with residue γ_i. This is the idea behind the algorithm and the proof of Proposition 2.2.

Proposition 2.2. *Let $a, b \in C(x)$. Then:*

1. $\int \frac{a}{b} dx = \sum_{i=1}^{n} \gamma_i \log(v_i)$.
2. *For every expression $\sum_{i=1}^{m} \delta_i \log(w_i)$ with $\delta_i \in \mathbb{C}$, $w_i \in \mathbb{C}(x)$, that is a solution of $y' = \frac{a}{b}$, one has*

$$\gamma_1, \dots, \gamma_n \in \mathbb{Q}(\delta_1, \dots, \delta_m).$$

Proof. Each v_i is square-free since $v_i \mid b$. For $i \neq j$ one has $\gcd(v_i, v_j) = \gcd(a - \gamma_i b', a - \gamma_j b', b) = 1$ since $\gcd(b', b) = 1$. Hence $v := v_1 \cdots v_n$ is a divisor of b. Let β be a zero of b. Then $b'(\beta) \neq 0$ and $\gcd(a - \frac{a(\beta)}{b'(\beta)} b', b) \neq 1$. Therefore $\frac{a(\beta)}{b'(\beta)}$ is some γ_i and β is also a zero of $v_i = \gcd(a - \gamma_i b', b)$. Since b is square-free, it follows that $v = b$.

We want to show that $\frac{a}{b} = \sum_{i=1}^{n} \gamma_i \frac{v_i'}{v_i}$. This is equivalent to proving that $P := a - \sum_i \gamma_i v_i' \frac{v}{v_i}$ is actually 0. The polynomial P has degree less than the degree of b. For any j the polynomial v_j is a divisor of $a - \gamma_j b' = a - \gamma_j \sum_{i=1}^{n} \frac{v_i'}{v_i} v$. Now

$$\gcd(v_j, P) = \gcd(v_j, a - \gamma_j v_j' \frac{v}{v_j}) = \gcd(v_j, a - \gamma_j \sum_{i=1}^{n} v_i' \frac{v}{v_i}) = v_j,$$

since v_j divides $v_i' \frac{v}{v_i}$ for $i \neq j$. Hence P is divisible by all v_j and hence by $v = v_1 \cdots v_n = b$. Since the degree of P is less than the degree of b, one has $P = 0$. This proves the first statement.

Let a formula $\frac{a}{b} = \sum \delta_i \frac{w_i'}{w_i}$ be given. We transform it by writing $w_i = c_i \frac{f_i}{g_i}$, with $f_i, g_i \in \mathbb{C}[x]$ monic polynomials and c_i constants. The result is $\frac{a}{b} = \sum \delta_i' \frac{\tilde{w}_i'}{\tilde{w}_i}$ with $\tilde{w}_i \in \mathbb{C}[x]$ monic. Then

$$\mathbb{Q}(\delta_1', \dots, \delta_*') \subset \mathbb{Q}(\delta_1, \dots, \delta_n).$$

In other words, we may already suppose that the initial w_i's are monic polynomials in $\mathbb{C}[x]$. In the next step every w_i is decomposed as a product of monic linear factors. We find an expression $\frac{a}{b} = \sum \delta_i'' \frac{(x-\alpha_i)'}{x-\alpha_i}$ and again $\mathbb{Q}(\delta_1'', \ldots) \subset \mathbb{Q}(\delta_1, \ldots, \delta_n)$. The terms in this sum with the same coefficient δ_i'' are collected and this gives the result $\frac{a}{b} = \sum_{i=1}^{m} \epsilon_i \frac{w_i'}{w_i}$, where the w_i are monic, mutually coprime and square-free, the ϵ_i are distinct and all $\epsilon_i \in \mathbb{Q}(\delta_1, \ldots, \delta_n)$. The proof will be finished when we can show that the last expression for $\frac{a}{b}$ is identical with $\sum_i \gamma_i \frac{v_i'}{v_i}$. That the expressions are identical (up to order) follows from $b = w_1 \cdots w_m$ and

$$\gcd(a - \epsilon_j b', b) = \gcd(\sum_i (\epsilon_i - \epsilon_j) \frac{w_i'}{w_i}(w_1 \cdots w_m), w_1 \cdots w_m) = w_j.$$

Indeed, every ϵ_j is equal to some $\gamma_{j'}$ and every w_j is equal to some $v_{j'}$.

Exercise 2.3. Find an expression for

$$\int \frac{2x+1}{x^4 + 2x^3 + x^2 - 2}\, dx$$

with the method of Proposition 2.2.

Hint: Use Maple for the calculation of the determinant $R(y)$. Let Maple find the solutions γ_i of $R(y) = 0$. Let Maple find the gcd's v_i.

Use the method of Proposition 2.2 to find an expression for

$$\int \frac{1}{x^3 + x + 1}\, dx.$$

Compare this with the solution found with the partial fraction decomposition of $\frac{1}{x^3+x+1}$.

3. The Equation $y' = fy$ with $f \in C(x)^*$

This time we want to know the solutions $y \neq 0$ which are algebraic over $C(x)$, i.e., $y \neq 0$ satisfies a polynomial equation $y^d + a_{d-1}y^{d-1} + \cdots + a_1 y + a_0 = 0$ with coefficients $a_i \in C(x)$.

Write $f = \frac{a}{b}$ with a, b polynomials, b monic and $\gcd(a, b) = 1$. We introduce a new variable y and form the polynomial $R(y) := R_x(a - yb', b) \in C[y]$. The answer to the question is:

Proposition 3.1. *The equation $y' = fy$ has an algebraic solution $\neq 0$ if and only if the following conditions are satisfied.*

1. $\gcd(b', b) = 1$ *and the degree of a is less than the degree of b.*
2. *All the zeros of $R(y)$ are rational numbers.*

Proof. Suppose that $y \neq 0$ is an algebraic solution of the equation $y' = fy$. Let the minimal polynomial equation of y over the field $C(x)$ be $y^d + a_{d-1}y^{d-1} + \cdots + a_1 y + a_0 = 0$. Differentiation of this equation yields

$$fdy^d + (fa_{d-1}(d-1) + a'_{d-1})y^{d-1} + \cdots + (fa_1 + a'_1)y + a'_0 = 0.$$

By minimality, this new equation for y must be a multiple of the equation of minimal degree d. This implies

$$a'_{d-1} = fa_{d-1},\ a'_{d-2} = 2fa_{d-1},\ \ldots, a'_1 = (d-1)fa_1,\ a'_0 = dfa_0.$$

The rational function a_0 is $\neq 0$. Write $a_0 = \prod_{i=1}^{s} v_i^{m_i}$, where the v_i are inequivalent irreducible polynomials and where the m_i are integers. Then $f = \sum_{i=1}^{s} \frac{(m_i/d)v_i'}{v_i} = \frac{a}{b}$ with $\gcd(a,b) = 1$. Clearly $b = \prod_{i=1}^{s} v_i$ and the degree of a is strictly less than the degree of b. Moreover $\gcd(b, b') = 1$. Furthermore $R(\lambda) = 0$ if and only if the $\gcd(a - \lambda b', b)$ is $\neq 1$. The last condition is equivalent to v_j divides $a - \lambda b'$ for some j. The term $a - \lambda b'$ is equal to $\sum_i (\frac{m_i}{d} - \lambda)\frac{b}{v_i}$. Clearly v_j divides $a - \lambda b'$ if and only if $\frac{m_j}{d} - \lambda = 0$. The zeros of $R(y)$ are therefore rational numbers. So f satisfies the conditions (1) and (2).

On the other hand, assume that f satisfies the two conditions. Let $\gamma_1, \ldots, \gamma_n$ denote the zeros of $R(y)$ and put $v_i = \gcd(a - \gamma_i b', b)$. According to Proposition 2.2, one has $f = \frac{a}{b} = \sum_{i=1}^{n} \gamma_i \frac{v_i'}{v_i}$. The γ_i are supposed to be rational numbers and as a consequence the v_i are polynomials with coefficients in C. Let N denote the common denominator of the γ_i. The expression $y = \prod_{i=1}^{n} v_i^{\gamma_i}$ is algebraic over $C(x)$, since $y^N \in C(x)$. It is clear that y is a solution of the equation $y' = fy$.

Exercise 3.2. Construct an algorithm for finding the algebraic solutions of the equation $y' = fy$ with $f \in \mathbb{Q}(x)^*$. Test whether $y' = fy$ has an algebraic solution and calculate the solution if there is one, for the following f's:

$$\frac{-97x^3 + 2x^2 + 129x + 6}{15x^4 + 30x^2 - 45} \quad \text{and} \quad \frac{7x^4 + 21x^2 + 4x + 10}{x^5 + 6x^3 + x^2 + 5x + 5}.$$

4. Rational Solutions of an Equation of Order n

For the differential equation

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_0 y = 0, \text{ with all } a_i \in C(x),$$

we try to find the solutions in $C(x)$.

It seems more general to study the solutions in $\overline{C}(x)$ of this equation. We will explain the theoretical reason why this gives no new information. Let $V \subset \overline{C}(x)$ denote the set of solutions of the equation. This is a vector space over $\overline{C}$ of finite dimension (see Lemmas 5.1 and 5.2). Let G denote the Galois group of the field extension $\overline{C} \supset C$. The group G acts on $\overline{C}(x)$ in the obvious way. This action commutes with the differentiation $'$ and so V is invariant under the action of G. For $g \in G$, $\lambda \in \overline{C}$ and $v \in V$ one has $g(\lambda v) = g(\lambda)g(v)$. It follows from [7, Chapter 10, Proposition 3], that there is a basis $v_1, \ldots, v_s$ of V over $\overline{C}$ such that the v_i are invariant under G and thus belong to $V \cap C(x)$. Hence it suffices to study the rational solutions in $C(x)$ of the equation above.

The idea of the algorithm is the following: Suppose that the denominator of a solution $y \in C(x)$ is known and that one also knows a bound on the degree of the numerator. Then the unknown coefficients of the numerator can be calculated with linear algebra.

A solution y can only have a pole at α if at least one of the a_i has a pole at α. Also, ∞ is a possible pole of y. Hence the location of the possible poles of y is known. What we have to do is to estimate the possible order of a pole of y.

The Algorithm

For notational convenience we restrict ourselves to $n = 3$. Again for notational convenience we start by investigating the order of the pole of y at 0. Suppose that the expansion of y at 0 is $x^s + *x^{s+1} + \cdots$ with $s < 0$. The expansion of a_i at 0 is written as $a_i = b_i x^{n_i} + *x^{n_i+1} + \cdots$, where the b_i are nonzero constants and the n_i are integers. In case $a_i = 0$, we put $n_i = \infty$ and b_i has no meaning. The four possible lowest powers of x in the equation $y^{(3)} + a_2 y^{(2)} + a_1 y^{(1)} + a_0 y$ are

$$s(s-1)(s-2)x^{s-3}, \ s(s-1)b_2 x^{s-2+n_2}, \ s b_1 x^{s-1+n_1}, \ b_0 x^{s+n_0}.$$

Let min denote the minimum of $-3, -2+n_2, -1+n_1, n_0$. The coefficient $I(s)$ of x^{s+min} can be written as

$$\epsilon_3 s(s-1)(s-2) + \epsilon_2 s(s-1)b_2 + \epsilon_1 s b_1 + \epsilon_0 b_0,$$

where $\epsilon_i = 1$ if the corresponding element in $\{-3, -2+n_2, -1+n_1, n_0\}$ is minimal and $\epsilon_i = 0$ otherwise. The expression for I is a nonzero polynomial in s (seen as a variable) of degree ≤ 3. Since $y^{(3)} + a_2 y^{(2)} + a_1 y^{(1)} + a_0 y = 0$, the coefficient $I(s)$ of x^{s+min} must be 0. Thus s is a solution of the equation $I(s) = 0$. The latter equation is often called the *indicial equation.* If there is no integer s with $I(s) = 0$ then we can stop the calculations since in that case there is no nonzero rational solution. If there is no negative integer s with $I(s) = 0$ but there is an integer $k \geq 0$ then we define $s_0 = 0$. If there is a negative integer solution of $I(s) = 0$ then $s_0 < 0$ denotes the smallest one.

We now return to the general equation of order 3. We will use the following notation: $\text{ord}_p(f)$ is the order of the function f at the point p.

Let $\alpha_1, \ldots, \alpha_r$ denote the poles of a_0, a_1, a_2. For every i, $1 \leq i \leq r$, the method above yields an integer $s_i \leq 0$ such that, for any rational solution

$y \neq 0$, one has $\mathrm{ord}_{\alpha_i}(y) \geq s_i$. (Or possibly we find that there are no rational solutions.) This means that we can write $y = \frac{T}{N}$ with known $N = \prod_i (x - \alpha_i)^{-s_i}$ and with some polynomial T. The next thing that we have to do is to estimate the degree of T. For this purpose we develop y, a_2, a_1, a_0 at ∞. The expansions have the form $y = x^t + *x^{t-1} + \cdots$ and $a_i = c_i x^{m_i} + *x^{m_i - 1} + \cdots$. If $a_i \neq 0$ then c_i is supposed to be a nonzero constant and m_i is an integer. If $a_i = 0$ then we put $m_i = -\infty$ and c_i has no meaning. The four possible highest powers of x in the equation $y^{(3)} + a_2 y^{(2)} + a_1 y^{(1)} + a_0$ are

$$t(t-1)(t-2)x^{t-3},\ t(t-1)c_2 x^{t-2+m_2},\ tc_1 x^{t-1+m_1},\ c_0 x^{t+m_0}.$$

Let max denote the maximum of $\{-3, -2+m_2, -1+m_1, m_0\}$. Let $J(t)$ be the expression

$$\epsilon_3 t(t-1)(t-2) + \epsilon_2 t(t-1)c_2 + \epsilon_1 t c_1 + \epsilon_0 c_0,$$

where the $\epsilon_i = 1$ if the corresponding term is equal to max and $\epsilon_i = 0$ otherwise. Then J is a nonzero polynomial of degree ≤ 3 in t (seen as a variable). If there is no integer t with $J(t) = 0$ then we stop the algorithm. In the other case, let s_∞ denote the largest integer that is a zero of J. Then we find that $t \leq s_\infty$. Expanding $y = \frac{T}{N}$ at infinity leads to the inequality degree(T) $\leq s_\infty$+degree(N). This is the bound that we are looking for.

Let d be the bound for the degree of T and write $T = t_d x^d + \cdots + t_0$. The equation satisfied by y produces a third order equation for T. This leads to a set of linear equations for the coefficients t_i. With linear algebra one can find all solutions. This ends the algorithm. In the following we discuss a variation on the above algorithm.

A Variation

We would like to work with this algorithm over the field $C = \mathbb{Q}$. There is now the problem that the poles of the a_i are algebraic numbers. Let α be an algebraic number with minimal polynomial $P = T^d + c_{d-1}T^{d-1} + \cdots + c_1 T + c_0 \in \mathbb{Q}[T]$ over $\mathbb{Q}$. The field $\mathbb{Q}(\alpha)$ is isomorphic to $\mathbb{Q}[T]/(P)$. The calculations in $\mathbb{Q}(\alpha)$ are in fact translated into calculations with polynomials modulo the ideal (P).

Algebraic numbers will slow down the computations and may cause other problems. How to avoid this?

We assume that we can factor polynomials over $\mathbb{Q}$ (cf. Chapter 4). Instead of working with an algebraic number which is a pole of some a_i, we will work with an irreducible monic factor $p \in \mathbb{Q}[x]$ of the denominator of some a_i. Every element in $\mathbb{Q}(x)$ has a unique expansion $\sum_{n \geq A} d_n p^n$ where the d_n are polynomials in $\mathbb{Q}[x]$ with degree less than the degree of p. We will use those expansions in order to find the number of factors p in the denominator of a solution y. Write $y = a(x)p^s + *p^{s+1} + \cdots$ with a a nonzero polynomial of degree less than the degree of p. Similarly, $a_i = b_i(x)p^{n_i} + *p^{n_i+1} + \cdots$. With notations similar to the ones used before, one finds an expression

$$I = \epsilon_3 s(s-1)(s-2)a(x)(p(x)')^3 + \epsilon_2 s(s-1)a(x)(p(x)')^2 a_2(x) +$$

$$\epsilon_1 s a(x) p(x)' a_1(x) + \epsilon_0 a(x) a_0(x) \bmod p.$$

This expression must be identically zero for the given s. Since $a(x)$ is invertible modulo p, we can omit $a(x)$. Let d be the degree of p. Then I can be written as $I = I_0 + I_1 x + \cdots + I_{d-1}x^{d-1}$ with all $I_j \in \mathbb{Q}[s]$ of degree ≤ 3. We are looking for integers s which are common zeros of the polynomials $I_0, \ldots, I_{d-1}$. If there is no such integer then there is no rational solution $y \neq 0$ of the equation. In that case we stop the algorithm. If there is such an integer and the smallest one is ≥ 0 then we put $s_p = 0$. If there are negative integers satisfying the equations, then the smallest one is by definition $s_p < 0$. Now any solution $y \neq 0$ has the from $\frac{T}{N}$ with $N = \prod p^{-s_p}$. The rest of the algorithm is unchanged.

Exercise 4.1. Calculate the solutions in $\mathbb{Q}(x)$ of the equation $y'' = ry$ with

$$r = \frac{6x^4 + 26x^3 + 42x^2 + 30x - 8}{(x+1)^2(x+5)(x^3-1)}.$$

Exercise 4.2. Compute the rational solutions of the equation

$$y^{(3)} - \frac{8x^2 - 63x - 27}{(24x+27)x} y^{(2)} + \frac{448x^2 + 1080x + 1080}{3(8x+9)^2 x} y^{(1)} - \frac{24}{(8x+9)^2 x} y = 0.$$

5. Some Differential Galois Theory

A linear differential equation of *order n* is an equation of the form:

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1 y^{(1)} + a_0 y = f,$$

where $f, a_0, \ldots, a_{n-1}$ are also functions. The equation is called *homogeneous* if $f = 0$ and *inhomogeneous* if $f \neq 0$. The main question is to find out whether some solution or all solutions of the equation can be written in a 'closed formula'. We have already seen some algorithms related to this question. One can expect closed formulas only when the functions $f, a_0, \ldots, a_{n-1}$ are very well specified. The methods are algebraic and the conditions on the $f, a_0, \ldots, a_{n-1}$ will also be algebraic in nature. As before, C will denote a field of characteristic 0 and $\overline{C}$ will denote an algebraic closure of this field. In applications C and $\overline{C}$ will be subfields of the field $\mathbb{C}$ of complex numbers.

Most of the time we will assume that $f, a_0, \ldots, a_{n-1}$ belong to $C(x)$, the field of rational functions over C, or to $C((x))$, the *field of formal Laurent series* over the field C, consisting of the expressions $\sum_{n \geq N} a_n x^n$, with $N \in \mathbb{Z}$ and all $a_n \in C$. The expressions are 'formal' and there is no condition on the convergence. One can add and multiply the elements of $C((x))$ in the obvious way. Every nonzero element has an inverse (e.g., the inverse of $1 + x$ is the Laurent series $\sum_{n\geq 0}(-1)^n x^n$). Thus $C((x))$ is indeed a field. The operation $' = \frac{d}{dx}$ is defined by $(\sum_{n\geq N} a_n x^n)' = \sum_{n\geq N} n a_n x^{n-1}$.

Those two fields are examples of differential fields. A *differential field* K is a field equipped with a differentiation $a \mapsto a'$. The differentiation is supposed to satisfy the following rules:

$$(a+b)' = a' + b' \text{ and } (ab)' = a'b + ab'.$$

The *field of constants* C of K is the subfield $\{a \in K \mid a' = 0\}$. We will assume that the characteristic of K is 0 and that the differentiation on K is not trivial. In that case $\mathbb{Q} \subset C$ and $C \neq K$.

How many solutions of a homogeneous differential equation of order n over a differential field K are there? Let us write $W = \{w \in K \mid w \text{ satisfies the equation}\}$. It is clear that W is a vector space over the field of constants C of K.

Lemma 5.1. *The dimension of W over C is at most n.*

Proof. A matrix differential equation over K has the form $v' = Av$, where A is an $n \times n$-matrix with coefficients in K and where v is a vector of length n.

The order n homogeneous equation

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1 y^{(1)} + a_0 y = 0$$

can be translated in the following matrix equation $v' = Av$:

$$\begin{pmatrix} y \\ y^{(1)} \\ y^{(2)} \\ . \\ . \\ y^{(n-1)} \end{pmatrix}' = \begin{pmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 1 \\ -a_0 & -a_1 & -a_2 & . & . & -a_{n-1} \end{pmatrix} \begin{pmatrix} y \\ y^{(1)} \\ y^{(2)} \\ . \\ . \\ y^{(n-1)} \end{pmatrix}.$$

In Lemma 5.2 below we will show that $V := \{v \in K^n \mid v' = Av\}$ is a vector space over C with dimension $\leq n$.

Now the subspace W is mapped to V by

$$w \mapsto \begin{pmatrix} w \\ w^{(1)} \\ w^{(2)} \\ . \\ . \\ w^{(n-1)} \end{pmatrix}.$$

This map is easily seen to be C-linear and bijective. The dimensions of W and V are equal and so the dimension of W is $\leq n$.

Lemma 5.2. *$V := \{v \in K^n \mid v' = Av\}$ is a vector space over C with dimension $\leq n$.*

Proof. This follows from the statement:

If $v_1, \dots, v_k \in V$ are linearly dependent over K then the vectors are linearly dependent over C.

The proof of this statement is given by induction on k. The case $k = 1$ is obvious. Suppose $k > 1$ and suppose that $v_1, \dots, v_k$ are linearly dependent over K. In a nontrivial relation $f_1v_1 + \cdots + f_kv_k = 0$ (with all $f_i \in K$) we may suppose that $f_1 = 1$. Apply the operation $v \mapsto v' - Av$ to the equation. The result is an equality $f_2'v_2 + \cdots + f_k'v_k = 0$. If the f_i' are 0 for $2 \leq i \leq k$, then all $f_i \in C$ and we have found a linear relation over C. If not all f_i' are 0, then the induction hypothesis can be applied.

Remark 5.3. One can show that any matrix equation $v' = Av$, with A an $n \times n$-matrix over K, is equivalent to a matrix equation coming from a homogeneous order n equation over K. This means that we can switch back and forth between matrix equations and order n linear equations.

5.1 Picard-Vessiot Theory

We make a small excursion into ordinary Galois theory in order to explain later the ideas of differential Galois theory.

Consider a field K of characteristic 0 and a polynomial $P \in K[T]$ of degree n. We assume for convenience that $\gcd(P, P') = 1$ (i.e., P has no multiple zeros in any field extension of K). In general P does not have n zeros in K. One defines a *splitting field* L of P over K as follows:

1. K is subfield of L.
2. P has n (distinct) zeros in L.
3. L is minimal with respect to (1) and (2).

Let $a_1, \dots, a_n \in L$ denote the n zeros of P. Then the last condition can be replaced by $L = K(a_1, \dots, a_n)$; in words, 'the field L is generated over K by $a_1, \dots, a_n$'.

This splitting field is unique up to a K-linear isomorphism. The *Galois group* $G = Gal(L/K)$ of L over K is the group of the K-linear field automorphisms of L. This group permutes the elements of $\{a_1, \dots, a_n\}$. Any $g \in G$ is determined by its action on $\{a_1, \dots, a_n\}$, and so G can be considered as a subgroup of the permutation group of $\{a_1, \dots, a_n\}$. A main result of Galois theory is the *Galois correspondence*:

> There is a 1-1 correspondence between the subgroups H of G and the subfields M of L which contain K. This correspondence is given by the two maps, which are each other's inverses,

$$\begin{aligned} H &\mapsto L^H = \text{the elements of } L \text{ invariant under} \\ &\quad\ \text{the action of all } h \in H, \\ M &\mapsto \mathrm{Gal}(L/M) \\ &\quad\ (\text{N.B. } L \text{ is also a Galois extension of } M). \end{aligned}$$

The structure of the Galois group G gives important information about the polynomial equation $P(T) = 0$. In particular, the group G is solvable if and only if the solutions of the equation $P(T) = 0$ can be written in a 'closed form'. This means that the solutions can be expressed in terms of ordinary elements of K by using the symbols $\sqrt[m]{\ }$ and (of course) the operations $+, \cdot, (\ )^{-1}$.

It may come as a surprise that Galois theory has a perfect analogue for linear differential equations over a differential field K. We will give the highlights of what is called *Picard-Vessiot theory.* In the following we make the assumption that *the field of constants C of K is algebraically closed.*

In general, a homogeneous linear differential equation of order n over the differential field K does not have 'all' its solutions in K itself, i.e., the vector space $W = \{w \in K \mid\ w \text{ satisfies the equation}\}$ has dimension $< n$ over C. One tries to find a 'minimal differential field extension' L of K such that $W_L = \{w \in L \mid\ w \text{ satisfies the equation}\}$ does have dimension n over C. The precise formulation is the following:

$L \supset K$ is called a *Picard-Vessiot field* for the equation if:

1. $K \subset L$ is an extension of differential fields.
2. L has C as field of constants.
3. W_L (as defined above) has dimension n over C.
4. L is minimal in the sense that, for a differential field M with $K \subset M \subset L$ and dimension of W_M over C equal to n, the equality $M = L$ holds.

One can show that a Picard-Vessiot field exists and that two Picard-Vessiot fields for the same equation are isomorphic differential field extensions of K. In the special case $K = C(x)$ it is rather easy to show the existence of a Picard-Vessiot extension. We will indicate a proof in Lemma 6.1.

The *differential Galois group* G of the given equation over K and with Picard-Vessiot field L is defined as the group of the automorphisms σ of the field L such that σ is the identity on K and σ commutes with the differentiation of L. This group G acts in a C-linear way on W_L. Indeed, since σ commutes with $'$, one has for $w \in W_L$ that the element $\sigma w \in L$ is also a solution of the equation and therefore lies in W_L. This action of G on W_L induces a group homomorphism $G \to \mathrm{Aut}_C(W_L) \cong \mathrm{GL}(n, C)$. This homomorphism is injective and its image is an *algebraic subgroup* of $\mathrm{GL}(n, C)$. In this way G obtains the structure of a linear algebraic group over C. We will denote the differential Galois group by $\mathrm{DGal}(L/K)$.

The Galois correspondence has the following analogue:

There is a bijection between the algebraic subgroups H of G and the differential fields M with $K \subset M \subset L$. This correspondence is given by the two maps, which are each other's inverses,

$$\begin{array}{lcl} H & \mapsto & L^H = \text{ the elements of } L \text{ fixed under the action of } H, \\ M & \mapsto & \mathrm{DGal}(L/M) \end{array}$$

(N.B. L is also a Picard-Vessiot extension of M).

Special cases of this correspondence are:

1. Put $G = \mathrm{DGal}(L/K)$. The set L^G, of the G-invariant elements of L is equal to K.
2. Let $H \subset G$ be a normal algebraic subgroup of G. Then G/H is again a linear algebraic group and G/H can be identified with $\mathrm{DGal}(L^H/K)$.
3. Let G^o denote the connected component of the identity in G. Then G^o is a normal (algebraic) subgroup of G and G/G^o is a finite group. The field L^{G^o} is a finite Galois extension of K with (ordinary) Galois group equal to G/G^o.
4. $L \supset K$ is a finite extension (equivalently, all the solutions of the differential equation are algebraic over K) if and only if G is a finite group.

The differential equation over K and the corresponding Picard-Vessiot extension $L \supset K$ is called *Liouvillian* if there exists a sequence of differential subfields $K = L_0 \subset L_1 \subset \cdots \subset L_{n-1} \subset L_n$ with $L = L_n$ and such that for every i the extension $L_i \subset L_{i+1}$ has one of the following three forms:

1. $L_i \subset L_{i+1}$ is a finite algebraic extension.
2. $L_{i+1} = L_i(t)$ with $t' = f \in L_i$.
3. $L_{i+1} = L_i(t)$ with $\frac{t'}{t} = f \in L_i^*$.

Loosely stated, the differential equation is Liouvillian if all the solutions can be obtained from K by adding algebraic functions, primitives $\int f dx$ and exponentials of primitives $e^{\int f dx}$. The Liouvillian solutions of a differential equation are what we have vaguely called 'solutions in closed form' or 'symbolic solutions'.

The structure of the differential Galois group $\mathrm{DGal}(L/K)$ of a differential equation is the key to our questions about symbolic solutions. Indeed, one has the following result:

Let $G = \mathrm{DGal}(L/K)$ denote the differential Galois group of a differential equation over K. The equation is Liouvillian if and only if G^o is a solvable linear algebraic group (i.e., after conjugation the group G^o is a subgroup of the group of upper triangular matrices in $\mathrm{GL}(n, C)$)

We will develop this point of view in detail for differential equations of order two.

Examples 5.4. (1) Consider the equation $y' = f$ over K with $f \in K$. We suppose that this equation does not have already a solution in K. The equation can be replaced by the equivalent homogeneous one $y'' - \frac{f'}{f}y' = 0$. We claim that the Picard-Vessiot extension of this equation is $L = K(t)$, where t is transcendental over K and $t' = f$. It suffices to show that the set of constants of $K(t)$ is again C. As an illustration we will give a proof of this.

Any element of $K(t)$ can be written as $\frac{a}{b}$ where $a, b \in K[t]$, b is monic and $\gcd(a, b) = 1$. Suppose that $(\frac{a}{b})' = 0$. Then $a'b = ab'$ and so b divides b'. Write $b = t^d + b_{d-1}t^{d-1} + \cdots + b_0$ with all $b_i \in K$. If b has degree > 0, then $b' = (df + b'_{d-1})t^{d-1} + \cdots + (fb_1 + b'_0)$. If $df + b'_{d-1} = 0$ then the equation $y' = f$ has a solution in K. This contradiction implies that the degree of b' is $d - 1$. This is not possible since b is supposed to divide b'. We conclude that $b = 1$ and $a' = 0$. If the degree of a is > 0 then a' cannot be zero. Thus $a \in K$ and $a \in C$ since $a' = 0$.

The differential Galois group of the equation consists of the automorphisms σ with $\sigma(t) = t + c$ for some $c \in C$. Indeed, $\sigma(t)' = \sigma(t') = \sigma(f) = f$ and $(\sigma(t) - t)' = 0$. The differential Galois group is therefore isomorphic to the linear algebraic group $\mathbf{G}_a = C$, which is called the *additive group*.

(2) The equation $y' = fy$ with $f \in K^*$. There are two possibilities here. The first possibility is that, for every integer $n \neq 0$, the equation $y' = nfy$ has no solution $\neq 0$ in K. Then the Picard-Vessiot field is $L = K(t)$ with t transcendental over K and $t' = ft$. The differential Galois group consists of the automorphisms σ of the form $\sigma(t) = ct$, where $c \in C^*$. In other words, the differential Galois group is the linear algebraic group $\mathbf{G}_m = C^*$ (which is called the *multiplicative group*).

The second possibility occurs if there exist integers $n \neq 0$ such that $y' = nfy$ has a nonzero solution in K. Let $m > 0$ be the smallest positive integer with this property and let $g \in K^*$ satisfy $g' = mfg$. Then the Picard-Vessiot field of the equation is $K(t)$ where t is algebraic over K with minimal equation $t^m - g = 0$. The differential Galois group consists of the automorphisms σ with $\sigma(t) = \zeta t$ and $\zeta \in C$ such that $\zeta^m = 1$. In other words, the differential Galois group is the algebraic subgroup of $\mathbf{G}_m = C^*$ consisting of the mth roots of unity.

We leave the verification of (2) as an exercise for the reader. We note the connection of this example with Proposition 3.1.

(3) The Airy equation $y'' = xy$ over $C(x)$ has differential Galois group $\mathrm{SL}(2, C)$. This will be proved in Section 8. The group is connected and not solvable. This has as consequence that the solutions (the Airy functions) cannot be obtained from the rational functions by means of integrals and exponentials of integrals and algebraic functions.

6. Order Two Equations Over $C(x)$

As usual, the algebraic closure of C will be denoted by $\overline{C}$. Consider the equation $y'' + ay' + by = 0$ with $a, b \in C(x)$. It has a certain differential Galois group $G \subset \mathrm{GL}(2, \overline{C})$.
One can transform the equation $y'' + ay' + by = 0$ into the equation $v'' = rv$, with $r \in C(x)$, by the substitution $y = fv$ where $f = \exp(-\frac{1}{2} \int a\, dx)$. The term r is equal to $\frac{1}{4}a^2 + \frac{1}{2}a' - b$. The new equation $v'' = rv$ has a differential Galois group H which is, according to the next lemma, an algebraic subgroup of $\mathrm{SL}(2, \overline{C})$. This simplifies matters.

However the transformation uses maybe an extension of the differential field $C(x)$, since the equation $\frac{f'}{f} = -\frac{1}{2}a$ need not have a solution in $C(x)$. As a consequence, solving $v'' = rv$ is not quite the same as solving $y'' + ay' + by = 0$. One can show that the two differential Galois groups are related by $G/Z \cong H/T$, where Z and T are the subgroups of G and H consisting of the multiples of the identity. In the second part of Remark 9.3, we will encounter an example of this situation.

This means that our restriction, in this section, to equations of the form $y'' = ry$ is a slight loss of generality.

Lemma 6.1. *There is a Picard-Vessiot field for the equation $y'' = ry$. The differential Galois group of the equation is a subgroup of* $\mathrm{SL}(2, \overline{C})$.

Proof. Let $d \in C$ be such that r has no pole at d. We introduce the variable $t = x - d$ and write $r = \sum_{n \geq 0} r_n t^n$. We try to find solutions of the form $y = \sum_{n \geq 0} a_n t^n$ of the equation $y'' = ry$. This leads to the set of equations

$$(n+2)(n+1)a_{n+2} = \sum_{i+j=n} r_i a_j \text{ for all } n \geq 0.$$

Here, a_0 and a_1 can be chosen arbitrarily. After this choice, the a_n are determined for $n \geq 2$. So we find two solutions $y_1 = 1 + 0 \cdot t + *t^2 + \cdots$ and $y_2 = t + *t^2 + \cdots$ which are linearly independent over $\overline{C}$. The subfield $K = \overline{C}(x)(y_1, y_1', y_2, y_2') \subset \overline{C}((x-d))$ is easily seen to be a Picard-Vessiot field for the equation. An element σ in the differential Galois group has, with respect to the basis y_1, y_2 of the space V of all solutions of the equation in K, the matrix $\begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}(2, \overline{C})$. In order to see that the determinant of this matrix is 1, we consider the expression $y_1 y_2' - y_1' y_2$. The derivative of this expression is 0 and so $y_1 y_2' - y_1' y_2 \in \overline{C}$. In particular, σ leaves this element invariant. As $\sigma(y_1 y_2' - y_1' y_2) = (ad - bc)(y_1 y_2' - y_1' y_2)$, we conclude $ad - bc = 1$.

In the next proposition we have collected the relevant information about the algebraic subgroups of $\mathrm{SL}(2, \overline{C})$.

Proposition 6.2. *The algebraic subgroups G of* $\mathrm{SL}(2,\overline{C})$ *are, up to conjugation in* $\mathrm{SL}(2,\overline{C})$, *classified as follows:*

1. $G = \mathrm{SL}(2,\overline{C})$.
2. *G is reducible, i.e., G is a subgroup of* $\left\{\begin{pmatrix} * & * \\ 0 & * \end{pmatrix}\right\}$.
3. *G is irreducible and imprimitive, i.e., G is a subgroup of the infinite dihedral group*
$$D_\infty := \left\{\begin{pmatrix} a & 0 \\ 0 & a^{-1} \end{pmatrix}\right\} \cup \left\{\begin{pmatrix} 0 & -a \\ a^{-1} & 0 \end{pmatrix}\right\}$$
and G is irreducible.
4. *G is a finite primitive group which means that G has the properties:*
   - ○ *G is finite.*
   - ○ *G is irreducible, i.e., no line in $\overline{C}^2$ is invariant under G.*
   - ○ *G is not imprimitive, i.e., there is no pair of lines in $\overline{C}^2$ such that G permutes these two lines.*

There are three finite primitive groups in $\mathrm{SL}(2,\overline{C})$ *(up to conjugation), they are called the tetrahedral group, the octahedral group, and the icosahedral group. The images of those groups in* $\mathrm{PSL}(2,\overline{C}) = \mathrm{SL}(2,\overline{C})/\{\pm\}$ *are isomorphic to the groups* A_4, S_4, *and* A_5. *The three groups have order* 24, 48, *and* 120, *respectively.*

Except for the three finite groups of part (4) of the proposition, there are the following representatives of the conjugacy classes of finite subgroups of $\mathrm{SL}(2,\overline{C})$:

- ○ The cyclic group of order n with generator $\begin{pmatrix} \zeta_n & 0 \\ 0 & \zeta_n^{-1} \end{pmatrix}$ with ζ_n a primitive n-th root of unity.
- ○ The finite dihedral groups D_m given as
$$D_m := \left\{ \begin{pmatrix} \zeta_{2m}^i & 0 \\ 0 & \zeta_{2m}^{-i} \end{pmatrix}, \begin{pmatrix} 0 & -\zeta_{2m}^i \\ \zeta_{2m}^{-i} & 0 \end{pmatrix} \;\middle|\; 0 \le i < 2m \right\},$$
where ζ_{2m} is a primitive $2m$th root of unity.

The Algorithm

The aim of the algorithm is to find, for a given equation $y'' = ry$ with $r \in C(x)$, the differential Galois group G and solutions in 'closed form' (if they exist). It turns out that there are no solutions in closed form if $G = \mathrm{SL}(2,\overline{C})$. If G is a proper subgroup of $\mathrm{SL}(2,\overline{C})$, then the Picard-Vessiot field of the equation turns out to be a Liouvillian extension of $C(x)$ and thus the solutions of $y'' = ry$ can be written in closed form.

The classification in Proposition 6.2 is the basis for the algorithm of Kovacic. We will explain some of the steps.

(1) Suppose that the equation $y'' = ry$ has a reducible differential Galois group G. Then there is a line $\overline{C}y \subset V = \overline{C}y_1 + \overline{C}y_2$ which is invariant under the action of G. For every $\sigma \in G$ there is a constant $c \in \overline{C}$ with $\sigma(y) = cy$. Then also $\sigma(y') = cy'$. Therefore $\sigma(\frac{y'}{y}) = \frac{y'}{y}$. The Galois correspondence of the Picard-Vessiot theory asserts that $u := \frac{y'}{y} \in \overline{C}(x)$. A small calculation shows that u satisfies the equation $u' + u^2 = r$. This equation is called the *Riccati equation* associated to $y'' = ry$. Thus a necessary condition for G to be reducible is that the Riccati equation $u' + u^2 = r$ has a solution in $\overline{C}(x)$. This is also sufficient! Indeed, let $u \in \overline{C}(x)$ satisfy $u' + u^2 = r$. Let $y \in K$, $y \neq 0$ be a solution of $y' = uy$. Then $y'' = ry$ and so $y \in V$. Let σ be an element of G. From $y' = uy$ and $u \in \overline{C}(x)$ it follows that $(\sigma y)' = u(\sigma y)$. Hence $\sigma(y) = cy$ for some $c \in \overline{C}^*$. And so the line $\overline{C}y \subset V$ is invariant under G and G is reducible.

(2) Suppose that the differential Galois group G is imprimitive. Then V has a basis, say again $\{y_1, y_2\}$, such that G permutes the lines $\overline{C}y_1$ and $\overline{C}y_2$. Then the elements $u_i := \frac{y_i'}{y_i}$, $i = 1, 2$, are also permuted by the elements of G. The elements $u_1 + u_2$ and u_1u_2 are invariant under G and belong therefore to $\overline{C}(x)$. The u_i are the zeros of the polynomial $T^2 - (u_1 + u_2)T + (u_1u_2) \in \overline{C}(x)[T]$. Hence the Riccati equation $u' + u^2 = r$ has a solution which is algebraic over $\overline{C}(x)$ of degree two. A continuation of this type of reasoning shows that *G is imprimitive if and only if the Riccati equation $u' + u^2 = r$ does not have a solution in $\overline{C}(x)$ and does have a solution which is algebraic over $\overline{C}(x)$ of degree two.*

(3) Similar arguments and a study of the finite primitive groups lead to the following statement.

Proposition 6.3. *Suppose that the differential Galois group of $y'' = ry$ is not equal to* $\mathrm{SL}(2, \overline{C})$. *Then there is a solution of the Riccati equation $u' + u^2 = r$ which is algebraic over $\overline{C}(x)$. Let $n \geq 1$ denote the minimal degree of such an algebraic solution of the Riccati equation. Then n can only have the values* $1, 2, 4, 6, 12$. *Furthermore:*

- *$n = 1$ if and only if G is reducible.*
- *$n = 2$ if and only if G is imprimitive.*
- *$n = 4$ if and only if G is conjugate to the tetrahedral group.*
- *$n = 6$ if and only if G is conjugate to the octahedral group.*
- *$n = 12$ if and only if G is conjugate to the icosahedral group.*

(4) The *procedure* for determining n and u (see [3] for more details) runs as follows:

(a) One starts by considering $n = 1$ and tries to compute a solution $u \in \overline{C}(x)$ of the Riccati equation. First one determines the singular points $d \in \overline{C}$ or ∞ of $y'' = ry$. For each singular point one calculates 'local solutions', i.e., working over the differential fields $\overline{C}((x - d))$ or $\overline{C}((x^{-1}))$, of the Riccati equation. For each singular point one finds at most two principal parts of a possible local solution. With some 'gluing' one tries to build a 'global solution', i.e., an element of $\overline{C}(x)$. This procedure is presented in more detail in the Sections 7, 8, and 9.

If this leads to a solution u, then one can reduce the equation $y'' = ry$ to order one (inhomogeneous) differential equations. A further study of those equations produces the differential Galois group and the closed form solutions.

(b) Suppose that no solution with $n = 1$ is found. Then one proceeds with the case $n = 2$. In principle the method of (a) can be copied, but applied to the second symmetric power of the equation $y'' = ry$. If a solution is found, then the equation $y'' = ry$ is reduced to order one (inhomogeneous) equations. The differential Galois group is then found as well as the solutions in closed form.

(c) One continues in a similar way with the cases $n = 4, 6, 12$ (in that order).

(d) If no algebraic solution u is found, then the differential Galois group is $\mathrm{SL}(2, \overline{C})$ and there are no solutions in closed form.

7. The Local Differential Galois Group

In this section we will (for convenience) assume that the field C is algebraically closed. We use the term 'local' in the sense that the differential field $C(x)$ is replaced by a larger differential field K. The field K is the completion of $C(x)$ at a point of $C \cup \{\infty\}$. This means that K is one of the fields $C((x - d))$ (with $d \in C$) or $C((x^{-1}))$.

The *local differential Galois group* of an equation over $C(x)$ with respect to one of the fields K is denoted by G_d or G_∞.

The local differential Galois group of an equation over $C(x)$ is easily determined. The well-known formal classification of differential equations (meaning the classification over the fields K), due to Turrittin, can be used to find this group.

Let a differential equation over $C(x)$ be given. Let G denote its differential Galois group. As we will see the local differential Galois groups G_d and G_∞ can be embedded as subgroups of G. The knowledge of the local differential Galois groups will be used in the sections 8 and 9 for the determination of G.

In this section we study the differential Galois group of the equation $y'' = ry$ over the field $C((x))$. The proof of Lemma 6.1 can be modified in order to show that this group is contained in $\mathrm{SL}(2, C)$.

For $r \in C((x))$, $r \neq 0$ one defines the *order* $\mathrm{ord}_0(r)$ of $r = \sum r_n x^n$ to be the minimal integer n with $r_n \neq 0$.

Proposition 7.1. *Let G_0 denote the differential Galois group of the equation $y'' = ry$ over the field $C((x))$. Write $r \in C((x))$, $r \neq 0$ as $r = \sum r_n x^n$. Then we have the following distinction of cases.*

1. $\mathrm{ord}_0(r) \geq 0$*; then* $G_0 = 1$.
2. $\mathrm{ord}_0(r) = -1$*; then* $G_0 = \mathbf{G}_a$.
3. $\mathrm{ord}_0(r) = -2$ *and* $r = r_{-2}x^{-2} + r_{-1}x^{-1} + \cdots$ *with* $r_{-2} \neq 0$. *Now*
   - *a)* $(1+4r_{-2})^{1/2} \notin \mathbb{Q}$*; then* $G_0 = \mathbf{G}_m$.
   - *b)* $(1+4r_{-2})^{1/2} \in \mathbb{Q} \setminus \mathbb{Z}$*; then G_0 is finite cyclic with order* ≥ 3.
   - *c)* $(1+4r_{-2})^{1/2} \in \mathbb{Z}$ *and odd; then G_0 is 1 or* $\mathbf{G}_a$.
   - *d)* $(1+4r_{-2})^{1/2} \in \mathbb{Z}$ *and even; then G_0 is $\{\pm 1\}$ or* $\{\pm 1\}\mathbf{G}_a$.
4. $\mathrm{ord}_0(r) \leq -3$ *and odd, then* $G_0 = D_\infty$.
5. $\mathrm{ord}_0(r) \leq -4$ *and even, then* $G_0 = \mathbf{G}_m$. .

If $(1+4r_{-2})^{1/2} = \frac{t}{n} \in \mathbb{Q} \setminus \mathbb{Z}$ with $\gcd(t,n) = 1$ and $n > 1$, then the order of G_0 is n if both t and n are odd. Otherwise the order of G_0 is $2n$.

Proof. The proof consists of somewhat long calculations with formal Laurent series. Let PV denote the Picard-Vessiot field of the equation and let V denote the solution space of the equation in the field PV. Then V has dimension two over C and the group G_0 lies in $\mathrm{SL}(V)$.

(1) If $\mathrm{ord}_0(r) \geq 0$ then the equation is regular. In particular, there are two solutions of the form $1 + *x + *x^2 + \cdots$ and $x + *x^2 + *x^3 + \cdots$ in $C((x))$. Thus $G_0 = 1$.

(2) Suppose $\mathrm{ord}_0(r) = -1$. Any solution in $C((x))$ can be normalized to $g = x^v + g_{v+1}x^{v+1} + g_{v+2}x^{v+2} + \cdots \in C((x))$. One finds $v = 1$ and recurrence relations for the coefficients g_i. Hence there is a solution $g \in C((x))$. Write $y = gF$. Then F, which we may suppose not to lie in C, satisfies the differential equation $F' = cg^{-2}$ with $c \in C^*$. This equation has no solution in $C((x))$ (otherwise $y'' = ry$ would have two independent solutions in $C((x))$). According to the first example of Section 5, the differential Galois group is $\mathbf{G}_a$.

(3) Suppose that $\mathrm{ord}_0(r) = -2$. We try to solve the associated Riccati equation $u' + u^2 = r$ with some $u \in C((x))$. The order of u at 0 is obviously -1. Write $u = u_{-1}x^{-1} + u_0 + u_1 x + u_2 x^2 + \cdots$. Then one finds a sequence of equations:

$$u_{-1}^2 - u_{-1} = r_{-2}, \tag{7.1}$$

$$2u_{-1}u_0 = r_{-1}, \tag{7.2}$$

$$(2u_{-1}+1)u_1 + u_0^2 = r_0 \tag{7.3}$$

$$\ldots\ldots \quad \ldots, \tag{7.4}$$

$$(2u_{-1}+(n+1))u_{n+1} + *** = r_n, \tag{7.5}$$

$$\ldots\ldots \quad \ldots \tag{7.6}$$

Then $u_{-1} = 1/2 \pm 1/2(1 + 4r_{-2})^{1/2}$. After a choice for u_{-1} the other coefficients of u are uniquely determined if moreover $2u_{-1} + n + 1$ is never zero. In the case that $2u_{-1} + n + 1 = 0$ for certain n there might be no solution of the Riccati equation starting with the chosen u_{-1} or there might be infinitely many solutions of Riccati starting with the chosen u_{-1}.

In case (a) there are two solutions, say u and $\tilde{u}$. They correspond with two lines L_1 and L_2, spanned by y_1, y_2, in the solution space V which are invariant under the group $G_0 \subset \mathrm{SL}(V)$. One has $u = \frac{y_1'}{y_1}$ and $\tilde{u} = \frac{y_2'}{y_2}$. Hence any $\sigma \in G_0$ has the form $\sigma(y_1) = cy_1$ and $\sigma(y_2) = c^{-1}y_2$ with $c \in C^*$. Hence $G_0 \subset \mathbf{G}_m$. Moreover $y_1 \in V$ is not algebraic over $C((x))$ since u_{-1} is not rational. Therefore, the equation $y' = uy$ has differential Galois group $\mathbf{G}_m$. Hence $G_0 = \mathbf{G}_m$.

Case (b) is similar to case (a). Now the differential Galois group of the equation $y' = uy$ is finite cyclic, since u_{-1} is rational. The order of this group, which is isomorphic to G_0, is equal to the smallest integer $N \geq 1$ such that Nu_{-1} is an integer. This proves case (b) and the last statement of the proposition.

In case (c) there is at least one solution u of the Riccati equation. Since u_{-1} is an integer, there is a $y_1 \in C((x))^*$ with $y_1' = uy_1$. Take a y_2 in V such that $\{y_1, y_2\}$ is a basis of V. Then any $\sigma \in G_0$ satisfies $\sigma(y_1) = y_1$ and $\sigma(y_2) = y_2 + cy_1$ with $c \in C$. Thus G_0 is an algebraic subgroup of the additive group $\mathbf{G}_a$. Hence G_0 is either 1 or $\mathbf{G}_a$.

Case (d) is similar to case (c). There is at least one solution u of the Riccati equation. Since $2u_{-1}$ is an odd integer, one finds that the solution $y_1 \neq 0$ of $y' = uy$ does not lie in $C((x))$ and $y_1^2 \in C((x))$. Choose $y_2 \in V$ such that $\{y_1, y_2\}$ is a basis. Any $\sigma \in G_0$ satisfies $\sigma(y_1) = \pm y_1$ and $\sigma(y_2) = \pm y_2 + cy_1$ with $c \in C$. Then G_0 is either equal to the group $\{\pm 1\}$ or is equal to $\{\pm 1\} \times \mathbf{G}_a$.

(4) Suppose that $\mathrm{ord}_0(r) = 2n+1$ with $n \leq -2$. Then the Riccati equation $u' + u^2 = r$ has two solutions u and $\tilde{u}$ in the field $C((x^{1/2}))$ (and no solution in $C((x))$ itself). The elements $y_1, y_2 \in V$ with $u = \frac{y_1'}{y_1}$ and $\tilde{u} = \frac{y_2'}{y_2}$ are not algebraic over $C((x^{1/2}))$ since u and $\tilde{u}$ have the form $*x^{n+1/2} + \cdots$ and $n + 1/2 \neq -1$. The differential Galois group of the equation over the field $C((x^{1/2}))$ consists of the σ with $\sigma y_1 = cy_1$ and $\sigma y_2 = c^{-1}y_2$ with any $c \in C^*$. The group G_0 contains this group and also contains an element that permutes the two lines Cy_1 and Cy_2. Hence $G_0 = D_\infty$.

(5) Suppose that $\mathrm{ord}_0(r)$ is even and ≤ -4. The two solutions of Riccati are in $C((x))$ and one easily sees that y_1, y_2 are not algebraic over $C((x))$. Hence $G_0 = \mathbf{G}_m$.

Remarks 7.2. (1) In the cases (3c) and (3d) there is an algorithm for determining G_0. The length of this algorithm depends on the integer $(1 + 4r_2)^{1/2}$.
(2) Instead of the field of formal Laurent series in x one can also consider an equation over the field of formal Laurent series in $x - c$ for some $c \in C$.

The statements are the same in this case. Also the case of the field of formal Laurent series in the variable x^{-1} is important. This field is denoted by $C((x^{-1}))$. This field is associated with the point ∞. The elements of this field are the expressions $\sum_{n \leq N} a_n x^n$ with all $a_n \in C$ and $N \in \mathbb{Z}$. One defines the order $\mathrm{ord}_\infty(f)$ of an element $f = \sum_{n \leq N} a_n x^n$ as the smallest integer k with $a_{-k} \neq 0$. The differentiation in the field $C((x^{-1}))$ is again $' = \frac{d}{dx}$. Proposition 6.2 is the analogue of 7.1 for $C((x^{-1}))$. The differential Galois group over the field $C((x^{-1}))$ will be denoted by G_∞.
(3) The results on the local differential Galois groups will be used in the following two sections. We will explain just how they are used.

Let a differential equation $Ly = 0$ be given over $C(x)$. Then there is a Picard-Vessiot field PV and a differential Galois group G associated to this. The same equation can also be considered over the field $C((x))$. The Picard-Vessiot field over $C((x))$ will be denoted by PV_0. The field $C(x)$ embeds in $C((x))$. Let V be the subspace of PV_0 consisting of all the solutions of $L(y) = 0$ in PV_0. Let $C(x)(V)$ denote the smallest differental subfield of PV_0 containing $C(x)$ and V. It is easily seen that $C(x)(V)$ is a Picard-Vessiot field for $L(y) = 0$ over $C(x)$. Hence PV and $C(x)(V)$ are isomorphic differential fields over $C(x)$. In other words, there exists an embedding $PV \to PV_0$. This embedding is not unique. For a fixed choice of the embedding one finds an injective homomorphism $G_0 \to G$. Indeed, any $\sigma \in G_0$ acts on V as a C-linear map (and is determined by this action). Hence σ leaves the subfield $C(x)(V)$ invariant, is $C(x)$-linear and commutes with $'$. Thus σ induces a differential automorphism of PV over $C(x)$, that belongs to G. The homomorphism $G_0 \to G$ is clearly injective. Another choice of the embedding $PV \to PV_0$ induces a homomorphism $G_0 \to G$ which is obtained from the first one by conjugation with an element of G.

The differential Galois group of the equation $L(y) = 0$ over the field $C((x-c))$ is denoted by G_c. There is an injective homomorphism $G_c \to G$. In a similar way one finds an injective homomorphism $G_\infty \to G$.

One can show that the group G is generated as an algebraic group by the images of the $G_c \to G$ (with $c \in C$) and $G_\infty \to G$. This gives some information about the group G, but in general not enough to determine G. The problem here is that those images are only known up to conjugation with an element in G.

Proposition 7.3. *Calculation of G_∞.*

1. $\mathrm{ord}_\infty(r) \geq 4$; *then* $G_\infty = 1$.
2. $\mathrm{ord}_\infty(r) = 3$; *then* $G_\infty = \mathbf{G}_a$.
3. $\mathrm{ord}_\infty(r) = 2$ *and* $r = r_{-2}x^{-2} + r_{-3}x^{-3} + \cdots$ *with* $r_{-2} \neq 0$.
   - *a)* $(1+4r_{-2})^{1/2} \notin \mathbb{Q}$; *then* $G_\infty = \mathbf{G}_m$.
   - *b)* $(1+4r_{-2})^{1/2} \in \mathbb{Q} \setminus \mathbb{Z}$; *then* G_∞ *is finite cyclic with order* ≥ 3.
   - *c)* $(1+4r_{-2})^{1/2} \in \mathbb{Z}$ *and odd; then* G_∞ *is* 1 *or* $\mathbf{G}_a$.
   - *d)* $(1+4r_{-2})^{1/2} \in \mathbb{Z}$ *and even; then* G_∞ *is* $\{\pm 1\}$ *or* $\{\pm 1\}\mathbf{G}_a$.

4. $\mathrm{ord}_\infty(r) \le 1$ *and odd; then* $G_\infty = D_\infty$.
5. $\mathrm{ord}_\infty(r) \le 0$ *and even; then* $G_\infty = \mathbf{G}_m$.

If $(1+4r_{-2})^{1/2} = \frac{t}{n} \in \mathbb{Q}\setminus\mathbb{Z}$ *with* $\gcd(t,n)=1$ *and* $n>1$, *then the order of* G_∞ *is* n *if both* t *and* n *are odd. Otherwise the order of* G_∞ *is* $2n$.

8. The Equation $y'' = ry$ with $r \in C[x]$, $r \neq 0$

The equation $y'' = ry$ has ∞ as only singular point. The algorithm that will be presented here is a simplification of Kovacic's algorithm. Still an interesting part of the Kovacic algorithm is needed for finding symbolic solutions. The algorithm is based on the following result:

Proposition 8.1. *Consider the equation* $y'' = ry$ *where* $r \in C[x]$ *is a nonzero polynomial of degree* n.

1. *If* $n=0$ *then the Riccati equation has two solutions and the differential Galois group is conjugate to the multiplicative group*
$$\mathbf{G}_m = \left\{ \begin{pmatrix} a & 0 \\ 0 & a^{-1} \end{pmatrix} \middle|\, a \in \overline{C}^* \right\}.$$
2. *If* $n>0$ *then the differential Galois group is conjugate to the Borel group*
$$B = \left\{ \begin{pmatrix} a & b \\ 0 & a^{-1} \end{pmatrix} \middle|\, a \in \overline{C}^*,\ b \in \overline{C} \right\}$$
if and only if the Riccati equation has a solution in $C(x)$. *If the Riccati equation has a solution then* n *is even.*
3. *The differential Galois group is* $\mathrm{SL}(2,\overline{C})$ *if there is no solution of the Riccati equation in* $C(x)$.

Proof. Let $K \supset \overline{C}(x)$ denote the Picard-Vessiot field of the equation $y'' = ry$. Let G denote the differential Galois group and let G^o denote the component of the identity of G. The finite extension $\overline{C}(x) \subset F := K^{G^o}$ has Galois group G/G^o. The equation is regular at any point $c \in \overline{C}$ and thus K can be embedded into the field $\overline{C}((x-c))$. Hence also F can be embedded into $\overline{C}((x-c))$. This means that the field extension $\overline{C}(x) \subset F$ can only be ramified above the point ∞. It is well known that a nontrivial extension of $\overline{C}(x)$ is ramified above at least two points. Consequently, $F = \overline{C}(x)$, and by Galois correspondence $G = G^o$. In view of the classification of the algebraic subgroups of $\mathrm{SL}(2,\overline{C})$, given in Section 6, one finds the following possibilities

$$\mathbf{SL(2)}, \boldsymbol{B}, \mathbf{G}_m, \text{ and } \mathbf{G}_a = \left\{ \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} \middle|\, b \in \overline{C} \right\}$$

for G. The local differential Galois group G_∞ is a subgroup of G. We have seen in Section 7 that G_∞ is either $\mathbf{G}_m$ for n even or D_∞ for n odd. This eliminates $\mathbf{G}_a$ as a candidate for G. Moreover, for odd n one has that $G = \mathrm{SL}(2, \overline{C})$.

The three groups $\mathrm{SL}(2, \overline{C}), B, \mathbf{G}_m$ occur if and only if the number of solutions of the Riccati equation is 0, 1, 2.

If $r \neq 0$ is a constant then the Riccati equation has the two solutions $\pm\sqrt{r}$. The differential Galois group is isomorphic to $\mathbf{G}_m$. The two symbolic solutions are $e^{\pm\sqrt{r}x}$.

If the degree of r is > 0 then we will show that the differential Galois group G cannot be isomorphic to $\mathbf{G}_m$. Suppose the contrary, then there are two solutions y_1, y_2 of $y'' = ry$ such that for any $\sigma \in G$ there is a $c \in \overline{C}$ with $\sigma(y_1) = cy_1$ and $\sigma(y_2) = c^{-1}y_2$. The element $f = y_1y_2$ is invariant under G and belongs to $\overline{C}(x)$ according to the Galois correspondence. A small calculation shows that f satisfies the equation $f^{(3)} - 4rf^{(1)} - 2r'f = 0$. We note that this equation is the *second symmetric power* of the equation $y'' = ry$. This third order equation has only a singular point at ∞. It follows (see Section 4) that the solution f must be a polynomial. We have therefore proved our claim if we can show that the operator $L : \overline{C}[x] \to \overline{C}[x]$, defined by $L(f) = f^{(3)} - 4rf^{(1)} - 2r'f$, has kernel 0. A small calculation shows that $L(x^k) = r_n(-4k - 2n)x^{n+k-1} + \cdots$ for $k \geq 0$. This implies that for a nonzero polynomial f of degree $k \geq 0$, the degree of $L(f)$ is $n + k - 1$. Hence L is injective.

We conclude that for $n > 0$ the group G can only be B or $\mathrm{SL}(2, \overline{C})$. The first case occurs if and only if the Riccati equation has a solution in $\overline{C}(x)$. We have already seen that n must be even in this case and that there is only one solution $u \in \overline{C}(x)$ of the Riccati equation. Hence this solution lies in $C(x)$.

The Algorithm

For nonconstant r we want to develop an algorithm that calculates the possible solutions in $\overline{C}(x)$ of the equation $u' + u^2 = r$. Suppose that u exists, then u has an expansion at ∞ of the form $a_nx^n + a_{n-1}x^{n-1} + \cdots \in \overline{C}((x^{-1}))$ with $a_n \neq 0$ and $n \geq 1$. It follows that the degree of r is $2n$. Hence we find that there are no solutions if the degree of r is odd.

Assume now that the degree of r is $2n > 0$. Then $u' + u^2$ has the expansion

$$a_n^2x^{2n} + (a_na_{n-1} + a_{n-1}a_n)x^{2n-1} + (a_na_{n-2} + a_{n-1}^2 + a_{n-2}a_n)x^{2n-2} + \cdots$$
$$+(a_na_{-1} + a_{n-1}a_0 + \cdots + a_{-1}a_n + na_n)x^{n-1} + \cdots$$

We only want to calculate the $a_n, \ldots, a_{-1}$. The equality $u' + u^2 = r$ leads to two solutions for $(a_n, \ldots, a_{-1})$.

On the other hand, let $\alpha \neq \infty$ be a pole of u. Then the equation $u' + u^2 = r$ proves that the expansion of u at α is equal to $u = \frac{1}{x-\alpha} + * + *(x - \alpha) + \cdots$. Let $\alpha_1, \ldots, \alpha_d$ denote the set of all the poles of u and put $F = \prod(x - \alpha_i)$. Then $u = v + \frac{F'}{F}$, where v is a polynomial in $\overline{C}[x]$. Hence $v = a_nx^n + \cdots + a_0$

and $d = a_{-1}$. A necessary condition is therefore that a_{-1} is an integer ≥ 0. Suppose that this condition is satisfied; then we proceed by calculating F. Write $F = x^d + f_{d-1}x^{d-1} + \cdots + f_1 x + f_0$. The equation $u' + u^2 = r$ leads to the equation $F'' + 2vF' + (v' + v^2 - r)F = 0$. Hence the coefficients of x^k, $k = 0, \ldots, n + d - 1$ in this expression are 0. This leads to a set of linear equations for the f_i. The coefficient of x^{n+d-1} can be seen to be identically 0. The equations for $k = n + d - 2, \ldots, n - 1$ determine $f_{d-1}, \ldots, f_0$ as one can see. Those $f_{d-1}, \ldots, f_0$ should also satisfy the equations for $k = 0, \ldots, n - 2$. If this is not the case, then there is no solution for the Riccati equation.

Example 8.2. The equation $y'' = (r_2 x^2 + r_1 x + r_0)y$ with $r_2 \neq 0$. Write $r_2 = A^2$ and $r_1 = 2AB$. The condition that $a_{-1} = d \in \mathbb{Z}$, $d \geq 0$ determines $r_0 = B^2 + (2d + 1)A$. (Note that we have supposed here (as we may) that $u = Ax + B + dx^{-1} + *x^{-2} + \cdots$.) The d linear equations for the coefficients of $F = x^d + f_{d-1}x^{d-1} + \cdots + f_0$ always have a unique solution. This can be seen as follows. Write $F = x^d + G$. The equation for G is $G'' + 2(Ax + B)G' - 2dAG = -2Bdx^{d-1} - d(d-1)x^{d-2}$. One considers the operator L on polynomials of degree $< d$, given by the formula $L(G) = G'' + 2(Ax + B)G' - 2dAG$. Then $L(x^k) = 2A(k - d)x^k +$ lower degree. Hence L is bijective and the equation $L(G) = -2Bdx^{d-1} - d(d-1)x^{d-2}$ has a unique solution.

The conclusion is that r of degree 2 gives the Borel group as differential Galois group if and only if r has the form $A^2x^2 + 2ABx + B^2 + (2d + 1)A$ for suitable $A \in \overline{C}^*$, $B \in \overline{C}$, $d \in \mathbb{Z}$, $d \geq 0$. (We note that actually $A, B \in C$.)

We continue with the case $d = 1$ in order to show how symbolic solutions can be calculated. The unique solution of the Riccati equation is $u = Ax + B + \frac{1}{x + A^{-1}B}$. A solution of the equation $y' = uy$ is

$$y_1 = (x + A^{-1}B)\exp(1/2Ax^2 + Bx).$$

Then $y_1'' = ry_1$. A second solution y_2 of the last equation can be obtained with the method of 'variation of constants'. Put $y_2 = fy_1$. Then f satisfies $f''y_1 + 2f'y_1' = 0$. A possibility for f' is y_1^{-2}. Hence $y_2 = y_1 \int y_1^{-2}\, dx$ is a second solution of the equation.

Exercise 8.3. Let $r = x^4 + 2x^3 + x^2 + 8x + e$. For which constants e does the Riccati equation $u' + u^2 = r$ have a solution $u \in \overline{\mathbb{Q}}(x)$?

Give symbolic expressions for two solutions of the equation $y'' = ry$ for the cases where u exists.

9. The Equation $y'' = ry$ with $r \in C[x, x^{-1}]$

In this section we develop some theory and an algorithm that determines the differential Galois group G and the possible Liouvillian solutions for this equation. As in Section 8, the algorithm is a simplification of Kovacic's algorithm for this special case. We note that there are at most two singular

points, namely 0 and ∞. The local differential Galois groups G_0 and G_∞ are almost completely determined by $\mathrm{ord}_0(r)$, $\mathrm{ord}_\infty(r)$ and r_{-2}. The groups G_0 and G_∞, which can be identified with certain subgroups of G, determine the possibilities for G. We start with the study of a special case.

Lemma 9.1. *For $r = r_{-2}x^{-2}$ (and $r_{-2} \neq 0$) one has:*

1. *If $r_{-2} = -1/4$ then $G = \{\pm 1\} \times \mathbf{G}_a$.*
2. *If $(1+4r_{-2})^{\frac{1}{2}}$ is equal to the positive rational number $\frac{t}{n}$ with $\gcd(t,n) = 1$, then G is a finite cyclic group of order n if both t and n are odd, and of order $2n$ if either t or n is even.*
3. *In all other cases $G = \mathbf{G}_m$.*

Proof. The Riccati equation $u' + u^2 = r$ has the solution(s) $u = u_{-1}x^{-1}$ with $u_{-1} = \frac{1}{2} \pm \frac{1}{2}(1+4r_{-2})^{1/2}$. For $r_{-2} = -1/4$ there is only one solution $\frac{1}{2}x^{-1}$ of the Riccati equation. Then $y_1 = x^{1/2}$ is a solution of $y'' = ry$. The second solution y_2 can be written as $fx^{1/2}$ for some (nonconstant) f. One finds that $f' = dx^{-1}$ with $d \in \overline{C}$, $d \neq 0$. This equation has no solution in $\overline{C}((x^{1/2}))$. It follows that the group G consists of the elements σ such that $\sigma(y_1) = \pm y_1$ and $\sigma(y_2) = \pm y_2 + cy_1$ with $c \in \overline{C}$. This proves case (1).

In the other cases there are at least two solutions of the Riccati equation and thus $G \subset \mathbf{G}_m$. One proves (2) and (3) as in Proposition 7.1.

Theorem 9.2. *Suppose that $r \neq r_{-2}x^{-2}$.*

1. *The table below gives the possibilities for the groups G_0, G_∞, G. The term C_4 indicates the cyclic group of order 4.*

	G_0	G_∞	G
1	$\neq C_4$	D_∞	SL(2)
2	D_∞	$\neq C_4$	SL(2)
3	C_4	D_∞	SL(2) *or* D_∞
4	D_∞	C_4	SL(2) *or* D_∞
5	$\neq D_\infty$	$\neq D_\infty$	$\mathbf{G}_m$, B *or* SL(2)

2. *In cases 3 and 4 (they are of course similar) the group G is equal to D_∞ if and only if the Riccati equation $u' + u^2 = r$ has a solution in $\overline{C}(x^{1/2})$.*
3. *In case 5, the group G is equal to $\mathbf{G}_m$, B or SL(2) if and only if the number of solutions of the Riccati equation in the field $\overline{C}(x)$ is 2, 1, or 0.*

Proof. (1) Suppose that $G = D_\infty$. We will show that either $G_0 = D_\infty$ and $G_\infty = C_4$, or $G_0 = C_4$ and $G_\infty = D_\infty$. This proves the cases 1, 2, 3, and 4 of the table.

Assume that $G = D_\infty$. Then $G^o \subset G$ has index two. Let K denote the Picard-Vessiot field of the equation. Then K^{G^o} is a quadratic extension of $\overline{C}(x)$. As in the proof of Proposition 8.1 one finds that this extension is only ramified above 0 and ∞. It follows that $K^{G^o} = \overline{C}(x^{1/2})$. The solution space $V \subset K$ of the equation has a basis y_1, y_2 such that the group G consists of all automorphisms with determinant 1, that permute the two lines Cy_1, Cy_2. The subgroup $G^o \subset G$ consists of the automorphisms σ such that $\sigma(y_1) = cy_1$ and $\sigma(y_2) = c^{-1}y_2$ with $c \in \overline{C}^*$. Then $u = \frac{y_1'}{y_1}$ and $\overline{u} = \frac{y_2'}{y_2}$ are invariant under G^o and therefore belong to $K^{G^o} = \overline{C}(x^{1/2})$. They are solutions of the Riccati equation and are conjugate over $\overline{C}(x)$. One can write those solutions as $u = A + Bx^{1/2}$ and $\overline{u} := A - Bx^{1/2}$ with $A, B \in \overline{C}(x)$ and $B \neq 0$. From

$$r = u' + u^2 = (A' + A^2 + xB^2) + (B' + x^{-1}B/2 + 2AB)x^{1/2},$$

one concludes that $A' + A^2 + xB^2 = r$ and $2AB + B' + x^{-1}B/2 = 0$. The last equation translates into $A = -\frac{1}{4}x^{-1} - \frac{1}{2}\frac{B'}{B}$.

Let $a \neq 0, \infty$ and suppose that B has a zero at a of order $k > 0$. The expansion of A at a is $A = \frac{-k/2}{x-a} + \cdots$ and the expansion of r at a reads $r = \frac{k/2}{(x-a)^2} + \frac{k^2/4}{(x-a)^2} + \cdots$. This produces the contradiction that r has a pole at a. We conclude that B has no zeros on C^*. In particular, $\text{ord}_0(B) + \text{ord}_\infty(B) \geq 0$. One has the following possibilities:

1. $k = \text{ord}_0(B) \geq -1$.
   The expansion of A at 0 is $A = (-\frac{1}{4} - \frac{1}{2}k)x^{-1} + * + \cdots$ and that of r is $r = (\frac{1}{4} + \frac{1}{2}k + (\frac{1}{4} + \frac{1}{2}k)^2)x^{-2} + *x^{-1} + \cdots$. One finds $(1 + 4r_{-2})^{1/2} = \frac{2k+3}{2}$ and Proposition 7.1 yields $G_0 = C_4$.
   Put $l = \text{ord}_\infty(B)$. The expansion of A at ∞ is $A = (-\frac{1}{4} + \frac{1}{2})x^{-1} + *x^{-2} + \cdots$ and the expansion of r at ∞ is
   $$r = ((\frac{1}{4} - \frac{l}{2}) + (\frac{1}{4} - \frac{l}{2})^2)x^{-2} + *x^{-3} + \cdots + *x^{1-2l} + \cdots.$$
   If $l \leq 1$ then $\text{ord}_\infty(r)$ is odd and ≤ 1. By Proposition 7.3 one has $G_\infty = D_\infty$.
   If $l \geq 2$ then $r = r_{-2}x^{-2} + *x^{-3} + \cdots$ and so $r = r_{-2}x^{-2}$. This case is excluded in the theorem.
2. $k = \text{ord}_0(B) \leq -2$ and so $l = \text{ord}_\infty(B) \geq 2$. Then $\text{ord}_0(r) = 2k - 1$ is odd and ≤ -3, hence $G_0 = D_\infty$. The expansion of r at ∞ and Proposition 7.3 imply that $G_\infty = C_4$.

We consider now case 5 and assume that $G_0 \neq D_\infty \neq G_\infty$. The group G is different from D_∞ as we have proved. Suppose that $G \subset \text{SL}(2, \overline{C})$ is a finite primitive group. Then the Picard-Vessiot field $K \supset \overline{C}(x)$ is a finite extension. As in the proof of Proposition 8.1 one shows that this extension is only ramified above the points 0 and ∞. The Galois group of $K/\overline{C}(x)$, which coincides with the differential Galois group, is then a finite cyclic group. This contradicts the assumption that G is a finite primitive group. Hence G is either SL(2) or a reducible group, i.e., contained in B.

We suppose now that $G \subset B$ and try to see that G can only be B or $\mathbf{G}_m$. It suffices to show that G contains $\mathbf{G}_m$, since in that case G can only be B or $\mathbf{G}_m$.

If $\mathrm{ord}_0(r) \geq 0$, then by Proposition 8.1 it follows that $G = B$ or $G = \mathbf{G}_m$.

If $\mathrm{ord}_0(r) = -1$, then $\mathrm{ord}_\infty(r) \leq 1$ and so $G_0 = \mathbf{G}_a$ and $G_\infty = \mathbf{G}_m$. Hence $G = B$.

If $\mathrm{ord}_0(r) = -2$, then $\mathrm{ord}_\infty(r) \leq 1$ since $r \neq r_{-2}x^{-2}$. Then $G_\infty = \mathbf{G}_m \subset G$.

If $\mathrm{ord}_0(r) \leq -3$, then $G_0 = \mathbf{G}_m \subset G$.

This finishes the verification.

(2) and (3) follow easily from the proof of (1) above.

The Algorithm

(1) The case $r = r_{-2}x^{-2}$ is completely described in Lemma 9.1.

(2) Determine which case of Theorem 9.2 occurs by using $\mathrm{ord}_0(r)$, $\mathrm{ord}_\infty(r)$, r_{-2}. In the cases 1, 2 the algorithm stops.

(3) Suppose that r satisfies case 5 of Theorem 9.2.

A possible solution of the Riccati equation has the form $v_0 + v_\infty + \frac{F'}{F}$ with $v_0 \in x^{-1}\overline{C}[x^{-1}]$, $v_\infty \in \overline{C}[x]$ and $F \in \overline{C}[x]$ a monic polynomial of degree d with $F(0) \neq 0$.

The term v_0 is given by:

1. If $\mathrm{ord}_0(r) \geq 0$, then $v_0 = 0$.
2. If $\mathrm{ord}_0(r) = -1$, then $v_0 = x^{-1}$.
3. If $\mathrm{ord}_0(r) = -2$, then $v_0 = (\frac{1}{2} \pm \frac{1}{2}(1+4r_{-2})^{1/2})x^{-1}$ (one or two possibilities).
4. If $\mathrm{ord}_0(r) = -2n \leq -4$, then $v_0 = *x^{-n} + \cdots + *x^{-1}$ such that $\mathrm{ord}_0(r - v_0^2 - v_0') \geq -n$ (two possibilities).

Let E denote the coefficient of x^{-1} in v_0.

The term $w_\infty = *x^{-1} + * + *x + \cdots + *x^m$ is defined by

1. If $\mathrm{ord}_\infty(r) \geq 3$, then $w_\infty = 0$.
2. If $\mathrm{ord}_\infty(r) = 2$, then $w_\infty = (\frac{1}{2} \pm \frac{1}{2}(1+4r_2)^{1/2})x^{-1}$ (one or two possibilities).
3. If $\mathrm{ord}_\infty(r) = -2m \leq 0$, then $w_\infty = *x^m + \cdots + * + *x^{-1}$ should satisfy $\mathrm{ord}_\infty(r - w_\infty^2 - w_\infty') \geq 2 - m$ (two possibilities).

Let D denote the coefficient of x^{-1} in w_∞. Put $v_\infty := w_\infty - Dx^{-1}$.

For a choice of the pair (v_0, w_∞) one calculates $D - E$. If this is not an integer and ≥ 0, then one tries another pair.

If a pair (v_0, w_∞) satisfies $D - E \in \mathbb{Z}$ and ≥ 0, then one considers a monic polynomial $F = x^d + f_{d-1}x^{d-1} + \cdots + f_0$ of degree $d = D - E$. This polynomial must satisfy the differential equation $F'' + 2vF' + (v' + v^2 - r)F = 0$ where $v = v_0 + v_\infty$. This leads to a set of linear equations for the f_i. If this has a solution, then u is found. If not, then one chooses another pair (v_0, w_∞).

(4) If r presents the case 3 or 4 of Theorem 9.2, then one has to solve the equation $u' + u^2 = r$ with $u \in \overline{C}(x^{1/2})$. This is done as in (3) above, with x replaced by $x^{1/2}$.

Remarks 9.3. 1. The calculation of the first coefficient of v_0 might involve a quadratic extension $\tilde{C}$ of the field C. If there is indeed a solution $u \in \overline{C}(x)$ (or in $\overline{C}(x^{1/2})$) of the Riccati equation, then there are two solutions. They have their coefficients in $\tilde{C}$ and are conjugated over C. The equation for (the first coefficient of) w_∞ must have a solution in $\tilde{C}$. The differential Galois group will be $G = \mathbf{G}_m$. Similar statements hold in case the calculation of w_∞ involves a quadratic extension of C.
2. Recent work of J.-P. Ramis, M.F. Singer and C. Mitschi on differential Galois theory describes and constructs differential equations over, say $\mathbb{C}(x)$ with a fixed number of singularities. In particular, Theorem 3 of the paper [6] describes all the differential Galois groups over the field of convergent Laurent series in the variable x. It is remarked ([6], page 267) that this coincides with the possibilities for the differential Galois groups of equations over $\mathbb{C}(x)$ having at most singular points at 0 and ∞. The subgroups of SL(2) in this list are

$$\mathrm{SL}(2),\ B,\ \mathbf{G}_m,\ \mathbf{G}_a,\ \{\pm 1\} \times \mathbf{G}_a,\ \text{the finite cyclic groups } C_n \text{ with } n \geq 1.$$

Why is the group $\mathbf{G}_a$ missing from our list?
Essentially the only equation for the group $\mathbf{G}_a$ is $y'' + x^{-1}y' = 0$. This equation is transformed by $y = zx^{-1/2}$ into $z'' = -\frac{1}{4}x^{-2}z$ which has differential Galois group $\{\pm 1\} \times \mathbf{G}_a$. The extra term $\{\pm 1\}$ obviously comes from the quadratic extension which is used in this transformation.

Exercise 9.4. Consider the equation $y'' = (r_{-1}x^{-1} + r_0 + r_1 x)y$ with $r_{-1} \neq 0$.

1. Prove that there only is a symbolic solution if $r_1 = 0$, $r_0 \neq 0$ and the Riccati equation $u' + u^2 = r_{-1}x^{-1} + r_0$ has a solution in $\overline{\mathbb{Q}}(x)$.
2. Suppose that u exists. Prove that the differential Galois group is equal to B. Prove that u is unique.
3. Find the first parts of the expansions of u at 0 and ∞.
4. Conclude that u has the form $u = x^{-1} + a + \frac{F'}{F}$, where a is a constant and where F is a monic polynomial of degree $d \geq 0$ with simple zeros such that $F(0) \neq 0$.
5. Conclude by looking at the coefficient of x^{-1} of the expansion of u at ∞ that $r_0 = (\frac{r_{-1}}{2(d+1)})^2$.
6. The condition $r_0 = (\frac{r_{-1}}{2(d+1)})^2$ with $d \in \mathbb{Z}$, $d \geq 0$ is necessary for the existence of u. Show that for $d = 0, 1$ the condition is also sufficient.
7. Calculate the symbolic solutions for the case $d = 0$.
8. Show that the condition is sufficient for any d by writing $F = x^d + f_{d-1}x^{d-1} + \cdots + f_0$ and by analyzing the system of linear equations that one obtains from

$$xF'' + (\frac{r_{-1}}{d+1}x + 2)F' - \frac{d}{d+1}r_{-1}F = 0.$$

Exercise 9.5. Calculate symbolic solutions for the equation

$$y'' = (x^{-3} - \frac{3}{16}x^{-2})y.$$

Hint: Find G_0 and G_∞. Calculate the 'negative part' v of the expansion at 0 for the possible solutions $u \in \overline{\mathbb{Q}}(x^{1/2})$. (The expansion of u at 0 is an element of $\overline{\mathbb{Q}}((x^{1/2}))$.) Conclude that u has the form $v + \frac{F'}{F}$ for some monic polynomial F in $x^{1/2}$. Find the degree of F.

Notes

The presentation of Sections 2 and 3 is influenced by a manuscript of A.H.M. Levelt on symbolic integration ([4]).

The algorithm of Section 8 is essentially the one invented by Liouville [5]. Our treatment can be seen as a modern version of Liouville's work.

The method and results of Section 9 are probably new. This section has its origin in an essay by B.E. Tuitman, written for her doctoral exam at the University of Groningen. Some of the exercises were composed by M. van Hoeij.

A good introduction to differential algebra is [1]. An extensive survey of the state of the art on differential Galois groups, algorithms for linear differential equations and the inverse problem for differential Galois theory is given in [8].

References

1. I. Kaplansky (1952): *An Introduction to Differential Algebra*, Hermann, Paris.
2. E. R. Kolchin (1973): *Differential Algebra and Algebraic Groups*, Academic Press, New York.
3. J. Kovacic (1986): *An algorithm for solving second order linear homogeneous differential equations*, J. of Symbolic Computation, 3–43.
4. A. H. M. Levelt (1992): *Lectures on symbolic integration*, University of Nijmegen.
5. J. Liouville (1834): *Mémoire, Sur l'intégration d'une classe d'équations différientielles du second ordre en quantités finies explicites*, Journal de Mathématiques pures et appliquées, 425–456.
6. J.-P. Ramis (1996): *About the inverse problem in differential Galois theory: The differential Abhyankar conjecture*, pp. 261–278 in: The Stokes Phenomenon and Hilbert's 16th Problem, (editors B.L.J. Braaksma, G.K Immink, M. van der Put), World Scientific, Singapore.
7. J.-P. Serre (1968): *Corps locaux*, Hermann, Paris.
8. M.F. Singer (1997): *Direct and Inverse Problems in Differential Galois Theory*, A survey for the "Collected works of Ellis Kolchin".

Chapter 10. Gröbner Bases for Codes

Mario de Boer and Ruud Pellikaan

1. Introduction

Coding theory deals with the following topics:

- ○ Cryptography or cryptology. Transmission of secret messages or electronic money, eavesdropping, intruders, authentication and privacy.
- ○ Source coding or data compression. Most data have redundant information, and can be compressed, to save space or to speed up the transmission.
- ○ Error-correcting codes. If the channel is noisy one adds redundant information in a clever way to correct a corrupted message.

In this and the following chapter we are concerned with Gröbner bases and error-correcting codes and their decoding. In Sections 2 and 3 a kaleidoscopic introduction is given to error-correcting codes centered around the question of finding the minimum distance and the weight enumerator of a code. Section 4 uses the theory of Gröbner bases to get all codewords of minimum weight of a cyclic code. Section 5 gives an elementary introduction to algebraic geometry codes.

All references and suggestions for further reading will be given in the notes at the end of this chapter. The beginning of this chapter is elementary and the level is gradually more demanding towards the end.

Notation: The ring of integers is denoted by $\mathbb{Z}$, the positive integers by $\mathbb{N}$ and the nonnegative integers by $\mathbb{N}_0$. The ring of integers modulo n is denoted by $\mathbb{Z}_n$. The number of elements of a set S is denoted by $\#S$. A field is denoted by $\mathbb{F}$ and its set of nonzero elements by $\mathbb{F}^*$. The finite field with q elements is denoted by $\mathbb{F}_q$. Vectors are row vectors. The transpose of a matrix M is written as $M^\top$. The inner product of the vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as $\mathbf{x}\cdot\mathbf{y} = \mathbf{x}\mathbf{y}^\top = \sum x_i y_i$. The projective space of dimension m over $\mathbb{F}_q$ is denoted by $PG(m, q)$. Variables are denoted in capitals such as $X, Y, Z, X_1, \ldots, X_m$. If I is an ideal and F an element of $\mathbb{F}_q[X_1, \ldots, X_m]$, then $\mathcal{Z}_{\mathbb{F}}(I)$ denotes the zero set of I in $\mathbb{F}^m$, and the coset of F modulo I is denoted by f.

2. Basic Facts from Coding Theory

Words have a fixed length n, and the letters are from an *alphabet* Q of q elements. Thus words are elements of Q^n. A *code* (dictionary) is a subset of Q^n. The elements of the code are called *codewords*.

2.1 Hamming Distance

Two distinct words of a code should differ as much as possible. To give this a precise meaning the *Hamming distance* between two words is introduced. If $\mathbf{x}, \mathbf{y} \in Q^n$, then

$$d(\mathbf{x}, \mathbf{y}) = \#\{i \mid x_i \neq y_i\}.$$

Exercise 2.1. Show that the Hamming distance is a *metric*. In particular, show that it satisfies the *triangle inequality*

$$d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}).$$

The *minimum distance* of a code C is defined as

$$d = d(C) = \min\{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in C,\ \mathbf{x} \neq \mathbf{y}\}.$$

2.2 Linear Codes

If the alphabet is a finite field, which is the case for instance when $Q = \{0, 1\}$, then Q^n is a vector space. A *linear code* is a linear subspace of $\mathbb{F}_q^n$. If a code is linear of dimension k, then the *encoding*

$$\mathcal{E} : \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n,$$

from message or source word $\mathbf{x} \in \mathbb{F}_q^k$ to encoded word $\mathbf{c} \in \mathbb{F}_q^n$ can be done efficiently by a matrix multiplication:

$$\mathbf{c} = \mathcal{E}(\mathbf{x}) = \mathbf{x}G,$$

where G is a $k \times n$ matrix with entries in $\mathbb{F}_q$. Such a matrix G is called a *generator matrix* of the code.

For a word $\mathbf{x} \in \mathbb{F}_q^n$ its *support* is defined as the set of nonzero coordinate positions, and its *weight* as the number of elements of its support, denoted by $\mathrm{wt}(\mathbf{x})$. The minimum distance of a linear code C is equal to its minimum weight

$$d(C) = \min\{\mathrm{wt}(\mathbf{c}) \mid \mathbf{c} \in C,\ \mathbf{c} \neq 0\}.$$

In this chapter a code will always be linear.

The parameters of a code C in $\mathbb{F}_q^n$ of dimension k and minimum distance d will be denoted by $[n, k, d]_q$ or $[n, k, d]$. Then $n - k$ is called the *redundancy*. For an $[n, k, d]$ code C we define the *dual code* $C^\perp$ as

$$C^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \mathbf{x} = 0 \ \text{ for all } \ \mathbf{c} \in C\}.$$

Exercise 2.2. Let C be a code of length n and dimension k. Show that $C^\perp$ has dimension $n - k$. Let H be a generator matrix for $C^\perp$. Prove that $C = \{\mathbf{c} \in \mathbb{F}_q^n \mid H\mathbf{c}^\top = 0\}$. Therefore H is called a *parity check* matrix for C.

Example 2.3. The $[7,4,3]$ *Hamming code* has generator matrix G and its dual, the $[7,3,4]$ *Simplex code* has generator matrix H, where

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Exercise 2.4. Let $(I_k|P)$ be a generator matrix of C, where I_k is the $k \times k$ identity matrix. Show that $(-P^{\top}|I_{n-k})$ is a parity check matrix of C.

2.3 Weight Distribution

Apart from the minimum distance, a code has other important invariants. One of these is the *weight distribution* $\{(i, \alpha_i) \mid i = 0, 1, \ldots, n\}$, where α_i denotes the number of codewords in C of weight i. The polynomials $W_C(X,Y)$ and $W_C(X)$, defined as

$$W_C(X,Y) = \sum_{i=0}^{n} \alpha_i X^{n-i} Y^i \quad \text{and} \quad W_C(X) = \sum_{i=0}^{n} \alpha_i X^{n-i}$$

are called the *(homogeneous) weight enumerators* of C. Although there is no apparent relation between the minimum distance of a code and its dual, the weight enumerators satisfy the *MacWilliams identity.*

Theorem 2.5. *Let C be an $[n,k]$ code over $\mathbb{F}_q$. Then*

$$W_{C^\perp}(X,Y) = q^{-k} W_C(X + (q-1)Y, X - Y).$$

2.4 Automorphisms and Isometries of Codes

Other important invariants of a code are its group of *automorphisms* and its group of *isometries.*

Let $Perm(n,q)$ be the subgroup of $GL(n,q)$ consisting of permutations of coordinates. Let $Diag(n,q)$ be the subgroup of $GL(n,q)$ consisting of diagonal matrices. Let $Iso(n,q)$ be the subgroup of $GL(n,q)$ which is generated by $Perm(n,q)$ and $Diag(n,q)$.

A code that is the image of C under an element of $Perm(n,q)$ is said to be *equivalent* to C. The subgroup of $Perm(n,q)$ that leaves C invariant is the *automorphism group* of C, $\mathrm{Aut}(C)$.

A code that is the image of C under an element of $Iso(n,q)$ is said to be *isometric* to C. The subgroup of $Iso(n,q)$ that leaves C invariant is the *isometry group* of C, $Iso(C)$.

Exercise 2.6. Show that $\mathrm{Aut}(C) = \mathrm{Aut}(C^{\perp})$ and similarly for $Iso(C)$.

Exercise 2.7. Show that a linear map $\varphi : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ is an isometry if and only if φ leaves the Hamming metric invariant, that is,

$$d(\varphi(\mathbf{x}), \varphi(\mathbf{y})) = d(\mathbf{x}, \mathbf{y}),$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$.

A code of length n is called *cyclic* if the cyclic permutation of coordinates $\sigma(i) = i - 1$ modulo n leaves the code invariant. See Section 4.

Exercise 2.8. Show that the [7, 4, 3] Hamming code, as defined in Example 2.3, is not cyclic, but that it is equivalent to a cyclic code.

3. Determining the Minimum Distance

Given a generator matrix of a code, the problem is to determine the minimum distance of the code. We will give five possible solutions here. All these methods do not have polynomial complexity in n, the length of the code. One cannot hope for a polynomial algorithm, since recently it has been proved that this problem is NP complete.

3.1 Exhaustive Search

This is the first approach that comes to mind. It is the brute force method: generate all codewords and check for their weights.

Since one generates the whole code, other invariants, like the weight distribution, are easy to determine at the same expence. But going through all codewords is the most inefficient way of dealing with the problem.

It is not necessary to consider all scalar multiples $\lambda\mathbf{c}$ of a codeword $\mathbf{c}$ and a nonzero $\lambda \in \mathbb{F}_q$, since they all have the same weight. This improves the complexity by a factor $q - 1$. One can speed up the procedure if one knows more about the code, for example the automorphism group, in particular for cyclic codes.

By the MacWilliams relations, given the weight distribution of a code, one can determine the weight distribution of the dual code by solving linear equations. Therefore it is good to do exhaustive search on whatever code (C or $C^{\perp}$) has lowest dimension.

Example 3.1. The Hamming code. Generating all 16 codewords of the Hamming code yields the following weight distribution of the code:

weight	# codewords
0	1
3	7
4	7
7	1

This could have been achieved by first computing the weight distribution of the dual code (dimension 3) and then applying the MacWilliams transform. Also, one can use that the code has a cyclic automorphism group of order 7. Therefore one knows that the number of codewords of weights 3 or 4 are multiples of 7.

Exercise 3.2. Does it hold in general that the number of codewords of a given weight in a cyclic code is a multiple of the length? If not, what is the exact relation?

3.2 Linear Algebra

In a sense the theory of linear codes is just 'linear algebra'. The determination of the minimum distance can be phrased in these terms as the following exercise shows.

Exercise 3.3. Show that the minimum distance is the minimal number of dependent columns in a parity check matrix.

But also for this method one has to look at all possible combinations of columns, and this number grows exponentially.

We give a sketch how the minimum distance of linear codes is determined by the algorithm of Brouwer. Let G be a $k \times n$ generator matrix of G. After a permutation of the columns and row reductions we may suppose that the first k columns form the $k \times k$ identity matrix. Any linear combination of w rows with nonzero coefficients gives a codeword of weight at least w. In particular, if the code has minimum distance 1, then we will notice this by the fact that one of the rows of G has weight 1. More generally, we look at all possible linear combinations of w rows for $w = 1, 2, \dots$ and keep track of the codeword of smallest weight. If we have found a codeword of weight v, then we can restrict the possible number of rows we have to consider to $v - 1$. The lower bound w for the weight of the codewords we generate is raised, and the lowest weight v of a codeword found in the process so far is lowered. Finally v and w meet.

An improvement of this method is obtained if G is of the form $(G_1 \cdots G_l)$ where $G_1, \dots, G_l$ are matrices such that the first k columns of G_j form the $k \times k$ identity matrix for all $j = 1, \dots, l$. In this way we know that any linear combination of w rows with nonzero coefficients gives a codeword of weight at least lw.

Exercise 3.4. Show that the maximum length of a binary code of dimension 4 and dual distance 3 is 7. What is the maximum length of a q-ary code of dimension k and dual distance 3? Hint: Use Exercise 3.3 and read the next section on finite geometry first.

3.3 Finite Geometry

It is possible to give the minimum distance of a code a geometric interpretation.

Suppose that the code is *nondegenerate*, this means that there is no coordinate j such that $c_j = 0$ for all codewords $\mathbf{c}$. For the determination of the minimum distance this is not an important restriction. So no column of the generator matrix G of a $[n, k, d]$ code is zero and its homogeneous coordinates can be considered as a point in projective space of dimension $k-1$ over $\mathbb{F}_q$. If two columns are dependent, then they give rise to the same point. In this way we get a set $\mathcal{P}$ of n points (counted with multiplicities) in $PG(k-1, q)$, which are not all contained in a hyperplane. This is called a *projective system.*

A projective system $\mathcal{P}$ of n points $P_1, \dots, P_n$ in $PG(k-1, q)$, with $P_j = (g_{1j} : \cdots : g_{kj})$, defines the code C with generator matrix $G = (g_{ij})$. This code depends on the choice of the enumeration of the points of $\mathcal{P}$ and on the choice of the homogeneous coordinates of P_j.

Two projective systems $\mathcal{P}_1$ and $\mathcal{P}_2$ are called equivalent if there exists a projective transformation $\sigma \in PGL(k-1, q)$ such that $\sigma(\mathcal{P}_1) = \mathcal{P}_2$.

Exercise 3.5. Show that in this way we get a one-to-one correspondence between isometry classes of nondegenerate $[n, k, d]$ codes and equivalence classes of projective systems of n points in $PG(k-1, q)$ such that the maximal number of points in a hyperplane (counted with multiplicities) is equal to $n - d$.

Example 3.6. The 7 columns of the $[7, 3, 4]$ Simplex code, viewed as homogeneous coordinates of points in $PG(2, 2)$, give the seven points of the *Fano plane.* All lines contain three points, so indeed the minimum distance is $7 - 3 = 4$.

Let $F(X, Y, Z) \in \mathbb{F}_q[X, Y, Z]$ be a homogeneous polynomial of degree m. Let $\mathcal{P}$ be the set of points $(a : b : c) \in PG(2, q)$ such that $F(a, b, c) = 0$, then we say that $\mathcal{P}$ is a *projective plane curve of degree* m in $PG(2, q)$ and that $F(X, Y, Z) = 0$ is its *defining equation.*

Exercise 3.7. What can be said about the minimum distance of the code of a plane curve in $PG(2, q)$ of degree m which has n points? Notice that the answer depends on whether the defining equation has a linear factor or not. Codes from plane curves are treated more extensively in Section 5.

Exercise 3.8. The *Klein quartic* is the projective plane curve with defining equation

$$X^3Y + Y^3Z + Z^3X = 0.$$

What are the parameters of the code associated to the Klein quartic over $\mathbb{F}_8$?

A *rational normal curve* in $PG(r, q)$ is the image of the map

$$\varphi : PG(1, q) \longrightarrow PG(r, q)$$

given by $\varphi(x_0 : x_1) = (x_0^r : x_0^{r-1}x_1 : \cdots : x_0x_1^{r-1} : x_1^r)$, or a projective transformation of this image.

Exercise 3.9. Show that the $q+1$ points of a rational normal curve in $PG(r,q)$ lie in *general linear position*, that is, no $r+1$ of these points lie in a hyperplane. What are the parameters of its associated code?

Exercise 3.10. Show that, possibly after a projective change of coordinates, the points of a rational normal curve are zeros of the 2×2 minors of the following matrix

$$\begin{pmatrix} X_0 & X_1 & \ldots & X_{r-1} \\ X_1 & X_2 & \ldots & X_r \end{pmatrix}.$$

What is the vanishing ideal of a rational normal curve in $PG(r,q)$?

Exercise 3.11. The *Hexacode* is the quaternary code with generator matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 1 & \alpha^2 & \alpha \end{pmatrix},$$

where $\alpha \in \mathbb{F}_4$ is a primitive element satisfying $\alpha^2 + \alpha + 1 = 0$. Show that the last 5 columns of G lie on the conic $X_0^2 + X_1X_2 = 0$ over $\mathbb{F}_4$, which is a rational normal curve. Use Exercise 3.9 to show that $d \geq 3$. Show that all 5 lines in $PG(2,4)$ through $(1:0:0)$, corresponding to the first column of G, intersect the remaining 5 points in exactly one point. Conclude that $d \geq 4$. Determine the weight distribution of the code using this geometric setting.

3.4 Arrangements of Hyperplanes

In this section we consider the dual picture.

Let C be a nondegenerate code. The columns of the generator matrix G can be considered as hyperplanes in $\mathbb{F}_q^k$ or $PG(k-1,q)$. Then column $\mathbf{g}_j^T$ corresponds to the hyperplane with equation $\sum_{i=1}^k g_{i,j}X_i = 0$. The multiset of hyperplanes will be denoted by $\mathcal{H}$.

Exercise 3.12. Show that the weight of a codeword $\mathbf{c} = \mathbf{x}G$ is given by

$$\mathrm{wt}(\mathbf{c}) = n - \text{number of hyperplanes in } \mathcal{H} \text{ through } \mathbf{x},$$

where this number is counted with multiplicities.

Clearly, the number of codewords of a certain weight t equals the number of points that are on exactly $n-t$ of the hyperplanes in $\mathcal{H}$. To find a nice expression for this we introduce the following notations. For a subset $J \subseteq \{1,2,\ldots,n\}$ we define

$$C(J) = \{\mathbf{c} \in C \mid c_j = 0 \text{ for all } j \in J\},$$

$$l(J) = \dim C(J).$$

Under the above correspondence we get the following isomorphism of vector spaces:

$$\bigcap_{j \in J} H_j \cong C(J).$$

Now define

$$\beta_t = \sum_{\#J=t} (q^{l(J)} - 1).$$

Exercise 3.13. Let $d^{\perp}$ denote the minimum distance of the dual code. Then show that for $t < d^{\perp}$

$$\beta_t = \binom{n}{t}(q^{k-t} - 1).$$

Exercise 3.14. Recall that α_s is the number of codewords of weight s. Prove the following formula,

$$\beta_t = \sum_{s=d}^{n-t} \binom{n-s}{t} \alpha_s,$$

by computing the number of elements of the set of pairs

$$\{(J, \mathbf{c}) \mid J \subseteq \{1, 2, \ldots, n\}, \#J = t, \mathbf{c} \in C(J),\ \mathbf{c} \neq 0\}$$

in two different ways.

Exercise 3.15. Show that the weight enumerator of C can be expressed in terms of the β_t as follows:

$$W_C(X) = X^n + \sum_{t=0}^{n-d} \beta_t (X-1)^t.$$

Exercise 3.16. Prove the following identity either by inverting the formula of Exercise 3.14 or by an inclusion/exclusion argument:

$$\alpha_s = \sum_{t=n-s}^{n-d} (-1)^{n+s+t} \binom{t}{n-s} \beta_t.$$

Example 3.17. The Hamming code, see Exercise 2.3. The seven hyperplanes in $\mathcal{H}$ are given by: $X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0, X_1 + X_2 + X_3 = 0, X_1 + X_2 + X_4 = 0, X_1 + X_3 + X_4 = 0$. Going through all points $\mathbf{x} \in \mathbb{F}_2^4$ and checking on how many of the hyperplanes in $\mathcal{H}$ they are, gives, after applying Proposition 3.15, the weight enumerator of the code. Computing the $l(J)$ for all J gives the following result:

$\#J$	0	1	2	3	4	4	5	6	7
$l(J)$	4	3	2	1	1 for 7 J	0 for 28 J	0	0	0

Since $\beta_i = 0$ for $i \geq 5$ and there are J of size 4 such that $C(J) \neq \{0\}$ we see that the minimum distance is $d = 3$. To find the weight distribution we compute the β_i.

β_0	$=$	$1(2^4 - 1) = 15$
β_1	$=$	$7(2^3 - 1) = 49$
β_2	$=$	$21(2^2 - 1) = 63$
β_3	$=$	$35(2^1 - 1) = 35$
β_4	$=$	$7(2^1 - 1) = 7$
β_5	$=$	0
β_6	$=$	0
β_7	$=$	0

α_0	$=$	1
α_1	$=$	0
α_2	$=$	0
α_3	$=$	$\beta_4 = 7$
α_4	$=$	$\beta_3 - 4\beta_4 = 7$
α_5	$=$	$\beta_2 - 3\beta_3 + 6\beta_4 = 0$
α_6	$=$	$\beta_1 - 2\beta_2 + 3\beta_3 - 4\beta_4 = 0$
α_7	$=$	$\beta_0 - \beta_1 + \beta_2 - \beta_3 + \beta_4 = 1$

Exercise 3.18. Compute the weight enumerator of the $[7, 3, 4]$ Simplex code and verify MacWilliam's identity.

Exercise 3.19. Compute the weight enumerator of the code on the Klein quartic of Exercise 3.8.

Exercise 3.20. Let C be an $[n, k, n-k+1]$ code. Show that $l(J) = n - \#J$ for all J. Compute the weight enumerator of such a code.

Exercise 3.21. Prove that the number $l(J)$ is the same for the codes C and $\mathbb{F}_{q^e}C$ in $\mathbb{F}_{q^e}^n$ for any extension $\mathbb{F}_{q^e}$ of $\mathbb{F}_q$.

Using the Exercises 3.14, 3.16 and 3.21 it is immediate to find the weight distribution of a code over any extension $\mathbb{F}_{q^e}$ if one knows the $l(J)$ over the ground field $\mathbb{F}_q$ for all subsets J of $\{1, \ldots, n\}$. Computing the $C(J)$ and $l(J)$ for a fixed J is just linear algebra. The large complexity for the computation of the weight enumerator and the minimum distance in this way stems from the exponential growth of the number of all possible subsets of $\{1, \ldots, n\}$.

Exercise 3.22. Let C be the code over $\mathbb{F}_q$, with q even, with generator matrix H of Exercise 2.3. For which q does this code contain a word of weight 7?

Exercise 3.23. Compare the complexity of the methods 'exhaustive search' and 'arrangements of hyperplanes' to compute the weight enumerator as a function of q and the parameters $[n, k, d]$ and $d^\perp$.

3.5 Algebra

Let the n hyperplanes in $\mathcal{H}$ have equations

$$L_1(X) = L_2(X) = \cdots = L_n(X) = 0,$$

where the L_i are linear forms in the variables $X_1, X_2, \ldots, X_k$ as in the previous section. Then a general codeword is of the form

$$\mathbf{c} = (L_1(\mathbf{x}), L_2(\mathbf{x}), \ldots, L_n(\mathbf{x})),$$

where $\mathbf{x} \in \mathbb{F}_q^k$. Now let I_t be the ideal generated by all products of t distinct $L_i(X)$, so

$$I_t = \left(\prod_{s=1}^{t} L_{i_s}(X) \mid 1 \leq i_1 < i_2 < \cdots < i_t \leq n \right).$$

If Φ_t is the ideal generated by all homogeneous forms of degree t in k variables $X_1, \ldots, X_k$, then clearly $I_t \subseteq \Phi_t$. We have the following.

Exercise 3.24. Show that

$$\mathcal{Z}_{\mathbb{F}_q}(I_t) = \{\mathbf{x} \in \mathbb{F}_q^k \mid \mathrm{wt}(\mathbf{c}) < t, \text{ with } \mathbf{c} = \mathbf{x}G\}.$$

Exercise 3.25. Show that

$$d = \min\{t \mid \mathcal{Z}_{\mathbb{F}_q}(I_{t+1}) \neq \{0\}\}.$$

Determining whether $\mathcal{Z}_{\mathbb{F}_q}(I_t) = \{0\}$ can be done by computing a Gröbner basis for the ideal. Sometimes it is easy to see that $I_t = \Phi_t$, whence one can conclude immediately that $\mathcal{Z}_{\mathbb{F}_q}(I_t) = \mathcal{Z}_{\mathbb{F}_q}(\Phi_t) = \{0\}$. But in general no polynomial algorithm is known to decide this question.

The ideals I_t are generated by $\binom{n}{t}$ elements and also this makes it infeasible to work with this method when n and t are large.

Contrary to what is done in exhaustive search, here all codewords are considered at once.

Example 3.26. The Hamming code, see Exercise 2.3. For this code we have

$$\mathbf{c} = (x_1, x_2, x_3, x_4, x_1 + x_2 + x_3, x_1 + x_2 + x_4, x_1 + x_3 + x_4).$$

It is easy to see that $I_1 = (X_1, X_2, X_3, X_4) = \Phi_1$ and hence $d \geq 1$. Also $I_2 = \Phi_2$ and $I_3 = \Phi_3$ is easy to check, so $d \geq 3$. To prove $d = 3$ it is enough to note that I_4 is contained in the ideal (X_1, X_2, X_3), so $(0,0,0,1) \in \mathcal{Z}_{\mathbb{F}_2}(I_4)$.

Example 3.27. For the Hexacode, see Exercise 3.11, it is easy to see that $I_1 = \Phi_1$ and $I_2 = \Phi_2$. We skip the computation of I_3 and compute a Gröbner basis for I_4. The result is:

$$\begin{aligned} I_4 \;=\; & (X_1^4, X_1^3X_2, X_1^3X_3, X_1^2X_2^2, X_1^2X_2X_3, X_1^2X_3^2, X_1X_2^3, X_1X_2^2X_3, \\ & X_1X_2X_3^2, X_1X_3^3, X_2^4, X_2^3X_3, X_2^2X_3^2, X_2X_3^3, X_3^4). \end{aligned}$$

We find that $I_4 = \Phi_4$ and hence $d \geq 4$. Since the rows of G are codewords of weight 4, we can conclude that $d = 4$. For completeness, a Gröbner basis is computed for I_5:

$$\begin{aligned} I_5 \;=\; & (X_1^4X_2 + X_1X_2^4, X_1^4X_3 + X_1X_3^4, X_1^3X_2X_3 + X_1X_2^2X_3^2, \\ & X_1^2X_2^2X_3 + X_1X_2X_3^3, X_1^2X_2X_3^2 + X_1X_2^3X_3, X_2^4X_3 + X_2X_3^4). \end{aligned}$$

Now I_5 is contained in the ideal (X_1, X_2), so $(0,0,1) \in \mathcal{Z}_{\mathbb{F}_4}(I_5)$ and indeed $d = 4$.

4. Cyclic Codes

In this section we consider a very important special class of codes: cyclic codes. We will find a nice algebraic description of the codewords of minimal weight in such codes and a way to decode up to half the minimum distance. It is not claimed that this is the most efficient way of treating this problem.

A *cyclic code* is a code C with the following property:

$$\text{if } \mathbf{c} = (c_0, c_1 \ldots, c_{n-1}) \in C, \quad \text{then } (c_{n-1}, c_0 \ldots, c_{n-2}) \in C.$$

In the context of cyclic codes it is convenient to consider the index i of a word as an element of $\mathbb{Z}_n$, the cyclic group of order n.

Consider the bijection ϕ between $\mathbb{F}_q^n$ and $\mathbb{F}_q[X]/(X^n - 1)$,

$$\phi(\mathbf{c}) = c_0 + c_1 X + \cdots + c_{n-1}X^{n-1}.$$

Then ideals in the ring $\mathbb{F}_q[X]/(X^n - 1)$ correspond one-to-one to cyclic codes in $\mathbb{F}_q^n$. In the rest of this chapter we will not distinguish between codewords and the corresponding polynomials under ϕ; we will talk about codewords $c(X)$ when in fact we mean the vector and vice versa.

Since $\mathbb{F}_q[X]/(X^n - 1)$ is a principal ideal ring, every cyclic code C is generated by a unique monic polynomial $g(X)$ of degree at most $n-1$, the *generator polynomial* $g(X)$:

$$C = \{c(X) \mid c(X) = r(X)g(X) \bmod (X^n - 1),\ r(X) \in \mathbb{F}_q[X]\}.$$

Instead of describing a cyclic code by its generator polynomial $g(X)$, one can describe the code by the set of zeros of $g(X)$ in an extension of $\mathbb{F}_q$.

From now on we assume that n is relatively prime with q. Let α be a primitive n-th root of unity in an extension field $\mathbb{F}_{q^e}$. A subset J of $\mathbb{Z}_n$ is called a *defining set* of a cyclic code C if

$$C = \{c(X) \in \mathbb{F}_q[X]/(X^n - 1) \mid c(\alpha^j) = 0 \text{ for all } j \in J\}.$$

The *complete defining set* $J(C)$ of C is defined as

$$J(C) = \{j \in \mathbb{Z}_n \mid c(\alpha^j) = 0 \text{ for all } c \in C\}.$$

Example 4.1. There are exactly two irreducible polynomials of degree 3 in $\mathbb{F}_2[X]$. They are factors of $X^7 + 1$:

$$X^7 + 1 = (X + 1)(X^3 + X + 1)(X^3 + X^2 + 1).$$

Let $\alpha \in \mathbb{F}_8$ be a zero of $X^3 + X + 1$. Then α is a primitive element of $\mathbb{F}_8$ and α^2 and α^4 are the remaining zeros of $X^3 + X + 1$. Consider the binary cyclic code C of length 7 with defining set $\{1\}$. Then $J(C) = \{1, 2, 4\}$ and $X^3 + X + 1$ is the generator polynomial of C. The code C is equivalent with the Hamming code.

Exercise 4.2. *BCH bound.* Show that a cyclic code has at least minimum distance d if $J(C)$ contains $d-1$ subsequent elements.

Exercise 4.3. The *cyclotomic coset* of $j \in \mathbb{Z}_n$ is the set $\{q^i j \mid i \in \mathbb{N}_0\}$. Show that a complete defining set is a union of cyclotomic cosets.

Exercise 4.4. Let C be a cyclic code of length 7 over $\mathbb{F}_q$. Show that $\{1,2,4\}$ is a complete defining set if q is even.

Exercise 4.5. Show that a binary cyclic code of length 11 has minimum distance 2 or 11.

Exercise 4.6. Show that the cyclotomic coset of $\{1\}$ in $\mathbb{Z}_{23}$ contains 4 subsequent elements for $q=2$.

4.1 The Mattson-Solomon Polynomial

Let $a(X)$ be a word in $\mathbb{F}_q^n$. Let $\alpha \in \mathbb{F}_{q^e}$ be a primitive n-th root of unity. Then the *Mattson-Solomon* (MS) polynomial of $a(X)$ is defined as

$$A(Z) = \sum_{i=1}^{n} A_i Z^{n-i}, \qquad A_i = a(\alpha^i) \in \mathbb{F}_{q^e}.$$

Here too we adopt the convention that the index i is an element of $\mathbb{Z}_n$, so $A_{n+i} = A_i$.

The MS polynomial $A(Z)$ is the *discrete Fourier transform* of the word $a(X)$. Notice that $A_n \in \mathbb{F}_q$.

Proposition 4.7.

1. *The inverse is given by* $a_j = \frac{1}{n} A(\alpha^j)$.
2. $A(z)$ *is the MS polynomial of a word* $a(X)$ *if and only if* $A_{jq} = A_j^q$ *for all* $j \in \mathbb{Z}_n$.
3. $A(z)$ *is the MS polynomial of a codeword* $a(X)$ *of the cyclic code* C *if and only if* $A_j = 0$ *for all* $j \in J(C)$ *and* $A_{jq} = A_j^q$ *for all* $j = 1, \ldots, n$.

Exercise 4.8. Let $\beta \in \mathbb{F}_{q^e}$ be a zero of $X^n - 1$. Show that

$$\sum_{i=1}^{n} \beta^i = \begin{cases} n & \text{if } \beta = 1 \\ 0 & \text{if } \beta \neq 1. \end{cases}$$

Expand $A(\alpha^i)$ using the definitions and use the above fact to prove Proposition 4.7(1). Prove the remaining assertions of Proposition 4.7.

Let $a(X)$ be a word of weight w. Then the *locators* $x_1, x_2, \ldots, x_w$ of $a(X)$ are defined as

$$\{x_1, x_2, \ldots, x_w\} = \{\alpha^i \mid a_i \neq 0\}.$$

Let $y_j = a_i$ if $x_j = \alpha^i$. Then

$$A_i = a(\alpha^i) = \sum_{j=1}^{w} y_j x_j^i.$$

Consider the product

$$\sigma(Z) = \prod_{j=1}^{w} (1 - x_j Z).$$

Then $\sigma(Z)$ has as zeros the reciprocals of the locators, and is (sometimes) called the *locator polynomial.* In this chapter and the following on decoding this name is reserved for the polynomial that has the locators as zeros.

Let $\sigma(Z) = \sum_{i=0}^{w} \sigma_i Z^i$. Then σ_i is the *i-th elementary symmetric function* in these locators:

$$\sigma_t = (-1)^t \sum_{1 \le j_1 < j_2 < \cdots < j_t \le w} x_{j_1} x_{j_2} \cdots x_{j_t}.$$

The following property of the MS polynomial is called the *generalized Newton identity* and gives the reason for these definitions.

Proposition 4.9. *For all i*

$$A_{i+w} + \sigma_1 A_{i+w-1} + \cdots + \sigma_w A_i = 0.$$

Exercise 4.10. Substitute $Z = 1/x_j$ in the equation

$$1 + \sigma_1 Z + \cdots + \sigma_w Z^w = \prod_{j=1}^{w} (1 - x_j Z)$$

and multiply by $y_j x_j^{i+w}$. This gives

$$y_j x_j^{i+w} + \sigma_1 y_j x_j^{i+w-1} + \cdots + \sigma_w y_j x_j^i = 0.$$

Check that summing over $j = 1, \ldots, w$ yields the desired result of Proposition 4.9.

Example 4.11. Let C be the cyclic code of length 5 over $\mathbb{F}_{16}$ with defining set $\{1,2\}$. Then this defining set is complete. The polynomial

$$X^4 + X^3 + X^2 + X + 1$$

is irreducible over $\mathbb{F}_2$. Let β be a zero of this polynomial in $\mathbb{F}_{16}$. Then the order of β is 5. The generator polynomial of C is

$$(X + \beta)(X + \beta^2) = X^2 + (\beta + \beta^2)X + \beta^3.$$

So $(\beta^3, \beta + \beta^2, 1, 0, 0) \in C$ and

$$(\beta + \beta^2 + \beta^3, 1 + \beta, 0, 1, 0) = (\beta + \beta^2)(\beta^3, \beta + \beta^2, 1, 0, 0) + (0, \beta^3, \beta + \beta^2, 1, 0)$$

is an element of C. These codewords together with their cyclic shifts and their nonzero scalar multiples give $(5+5)*15 = 150$ words of weight 3.

Using Propositions 4.7 and 4.9 it will be shown that these are the only codewords of weight 3. Consider the set of equations:

$$\left\{\begin{array}{ccccccccc} A_4 & + & \sigma_1 A_3 & + & \sigma_2 A_2 & + & \sigma_3 A_1 & = & 0 \\ A_5 & + & \sigma_1 A_4 & + & \sigma_2 A_3 & + & \sigma_3 A_2 & = & 0 \\ A_1 & + & \sigma_1 A_5 & + & \sigma_2 A_4 & + & \sigma_3 A_3 & = & 0 \\ A_2 & + & \sigma_1 A_1 & + & \sigma_2 A_5 & + & \sigma_3 A_4 & = & 0 \\ A_3 & + & \sigma_1 A_2 & + & \sigma_2 A_1 & + & \sigma_3 A_5 & = & 0. \end{array}\right.$$

If A_1, A_2, A_3, A_4 and A_5 are the coefficients of the MS polynomial of a codeword, then $A_1 = A_2 = 0$. If $A_3 = 0$, then $A_i = 0$ for all i. So we may assume that $A_3 \neq 0$. The above equations imply $A_4 = \sigma_1 A_3$, $A_5 = (\sigma_1^2 + \sigma_2)A_3$ and

$$\left\{\begin{array}{lcl} \sigma_1^3 + \sigma_3 & = & 0 \\ \sigma_1^2\sigma_2 + \sigma_2^2 + \sigma_1\sigma_3 & = & 0 \\ \sigma_1^2\sigma_3 + \sigma_2\sigma_3 + 1 & = & 0. \end{array}\right.$$

Substitution of $\sigma_3 = \sigma_1^3$ in the remaining equations yields

$$\left\{\begin{array}{lcl} \sigma_1^4 + \sigma_1^2\sigma_2 + \sigma_2^2 & = & 0 \\ \sigma_1^5 + \sigma_1^3\sigma_2 + 1 & = & 0. \end{array}\right.$$

Multiplying the first equation by σ_1 and adding the result to the second one gives

$$1 + \sigma_1\sigma_2^2 = 0.$$

Thus $\sigma_1 = \sigma_2^{-2}$ and

$$\sigma_2^{10} + \sigma_2^5 + 1 = 0.$$

This last equation has 10 solutions in $\mathbb{F}_{16}$, and we are free to choose A_3 from $\mathbb{F}_{16}^*$. This gives in total 150 solutions.

Exercise 4.12. Let C be the code of the previous example. Compute the number of codewords of weight 3 with the help of Exercise 3.20.

Exercise 4.13. Let C be a cyclic code of length 7 over $\mathbb{F}_q$ with defining set $\{1,2,4\}$. Show that $d(C) > 3$ if q is odd.

4.2 Codewords of Minimal Weight

The following way to get all minimal codewords of cyclic codes uses the theory of Gröbner bases.

Let C be a cyclic code of length n over $\mathbb{F}_q$ with defining set $J(C)$. Let $\mathbb{F}_{q^e}$ be an extension of $\mathbb{F}_q$ that contains an n-th root of unity. Let $S_C(w)$ be the following system of equations:

$$\begin{cases} A_{w+1} + \sigma_1 A_w + \cdots + \sigma_w A_1 = 0 \\ A_{w+2} + \sigma_1 A_{w+1} + \cdots + \sigma_w A_2 = 0 \\ \vdots \\ A_{w+n} + \sigma_1 A_{w+n-1} + \cdots + \sigma_w A_n = 0 \\ \text{for all } j \in J(C) \quad A_j = 0 \\ \text{for all } j \in \mathbb{Z}_n \quad A_{qj} = A_j^q. \end{cases}$$

In this system both the A_i and the σ_i are indeterminates.

From the properties of the MS polynomial stated in Propositions 4.7 and 4.9 we see that codewords of weight at most w give solutions of the system $\mathcal{S}_C(w)$, and that conversely any solution to the system comes from a codeword of weight at most w. The exact relation is as follows.

Theorem 4.14. *The solutions $(A_0, A_1, \ldots, A_{n-1})$ to $\mathcal{S}_C(w)$ over $\mathbb{F}_{q^e}$ are the coefficients of the MS polynomials of codewords of weight at most w.*

Corollary 4.15. *The minimum distance d is equal to the smallest value of w such that $\mathcal{S}_C(w)$ has a nonzero solution over $\mathbb{F}_{q^e}$. Each solution $(A_0, A_1, \ldots, A_{n-1})$ to $\mathcal{S}_C(d)$ over $\mathbb{F}_{q^e}$ corresponds one-to-one to a codeword of minimal weight.*

We conclude that the codewords of minimal weight in a cyclic code can be determined by solving a system of equations in the polynomial ring $\mathbb{F}_q[A_0, A_1, \ldots, A_{n-1}, \sigma_0, \sigma_1, \ldots, \sigma_d]$. Solving the system can be done by computing a Gröbner basis for the ideal defined by $\mathcal{S}_C(d)$. This method will be applied to the ternary Golay code in Project 7.

Exercise 4.16. Let C be a cyclic code of length 7 over $\mathbb{F}_q$, q even, with defining set $\{1, 2, 4\}$. Show that the number of codewords of weight 3 is equal to $7(q-1)$.

5. Codes from Varieties

Consider a geometric object $\mathcal{X}$ with a subset $\mathcal{P}$ consisting of n distinct points which are listed $P_1, \ldots, P_n$. Suppose that we have a vector space L over $\mathbb{F}_q$ of functions on $\mathcal{X}$ with values in $\mathbb{F}_q$. Thus $f(P_i) \in \mathbb{F}_q$ for all i and $f \in L$. In this way one has an evaluation map

$$ev_{\mathcal{P}} : L \longrightarrow \mathbb{F}_q^n$$

defined by $ev_{\mathcal{P}}(f) = (f(P_1), \ldots, f(P_n))$. If this evaluation map is linear, then its image is a linear code.

In the following, $\mathcal{X}$ is a subset of an *affine variety*, that is, the common set of zeros in affine space of some given set of polynomials. The points $P_1, \ldots, P_n$ are called ***rational*** when they have coordinates in $\mathbb{F}_q$. The functions will be polynomial functions.

Extending a reduction order on the set of monomials to a function on all polynomials gives an example of an order function. A special kind of order function is a weight function. The theory of Gröbner bases is used to show the existence of certain weight functions.

These order functions will be used to define codes and to derive a bound for the minimum distance for these codes that is similar to the BCH bound for cyclic codes.

5.1 Order and Weight Functions

Let $\mathbb{F}$ be a field. In this chapter an $\mathbb{F}$-algebra is a commutative ring with a unit that contains $\mathbb{F}$ as a unitary subring. Let R be an $\mathbb{F}$-algebra. An *order function* on R is a map

$$\rho : R \longrightarrow \mathbb{N}_0 \cup \{-\infty\},$$

that satisfies for $f, g, h \in R$ the following conditions:

$(O.0)$ $\rho(f) = -\infty$ if and only if $f = 0$.
$(O.1)$ $\rho(\lambda f) = \rho(f)$ for all nonzero $\lambda \in \mathbb{F}$.
$(O.2)$ $\rho(f+g) \leq \max\{\rho(f), \rho(g)\}$
and equality holds when $\rho(f) < \rho(g)$
$(O.3)$ If $\rho(f) < \rho(g)$ and $h \neq 0$, then $\rho(fh) < \rho(gh)$.
$(O.4)$ If $\rho(f) = \rho(g)$, then there exists a nonzero $\lambda \in \mathbb{F}$ such that $\rho(f - \lambda g) < \rho(g)$.

Here $-\infty < n$ for all $n \in \mathbb{N}_0$.

Example 5.1. Let $R = \mathbb{F}[X_1, \ldots, X_m]$. Let $\prec$ be a reduction order on the monomials in $X_1, \ldots, X_m$ that is isomorphic to the ordinary order on $\mathbb{N}$. The lexicographical total degree order is isomorphic with $(\mathbb{N}, <)$, but the lexicographical order is not if $m > 1$. Let the sequence $(F_i \mid i \in \mathbb{N})$ be an enumeration of the monomials in increasing order, so $F_i \prec F_{i+1}$ for all i. They form a basis of R over $\mathbb{F}$. So every nonzero polynomial F has a unique representation

$$F = \sum_{i \leq j} \lambda_i F_i,$$

where $\lambda_i \in \mathbb{F}$ and $\lambda_j \neq 0$. Define $\rho(F) = j - 1$. Then ρ is an order function on R.

Exercise 5.2. Let R be an $\mathbb{F}$-algebra. Show that there exists a sequence $(f_i \mid i \in \mathbb{N})$ which is a basis of R over $\mathbb{F}$ such that $\rho(f_i) < \rho(f_{i+1})$.

Let $L(l)$ be the vector space with basis $f_1, \ldots, f_l$. Let $l(i,j)$ be the smallest l such that $f_i f_j \in L(l)$. **Prove that $l(i,j)$ is strictly increasing in both arguments. Such a sequence is called *well-behaving*.**

Let R be an $\mathbb{F}$-algebra. A *weight function* on R is an order function on R that satisfies furthermore

$$(O.5) \quad \rho(fg) = \rho(f) + \rho(g)$$

for all $f, g \in R$. Here $-\infty + n = -\infty$ for all $n \in \mathbb{N}_0$.

If ρ is a weight function and $\rho(f)$ is divisible by an integer $d > 1$ for all $f \in R$, then $\rho(f)/d$ is again a weight function. Thus we may assume that the greatest common divisor of the integers $\rho(f)$ with $0 \neq f \in R$ is 1.

A *degree function* on R is a map that satisfies conditions $(O.0)$, $(O.1)$, $(O.2)$ and $(O.5)$. It is clear that condition $(O.3)$ is a consequence of $(O.5)$.

Example 5.3. The standard example of an $\mathbb{F}$-algebra R with a degree function ρ is obtained by taking $R = \mathbb{F}[X_1, \ldots, X_m]$ and $\rho(F) = \deg(F)$, the degree of $F \in R$. It is a weight function if and only if $m = 1$.

Let $\mathbf{w} = (w_1, \ldots, w_m)$ be an m-tuple of positive integers called *weights*. The *weighted degree* of $\alpha \in \mathbb{N}_0^m$ and the corresponding monomial X^α is defined as

$$\mathrm{wd}(X^\alpha) = \mathrm{wd}(\alpha) = \sum \alpha_l w_l,$$

and of a nonzero polynomial $F = \sum \lambda_\alpha X^\alpha$ as

$$\mathrm{wd}(F) = \max\{ \mathrm{wd}(X^\alpha) \mid \lambda_\alpha \neq 0 \}.$$

The *lexicographical total weighted degree order* $\prec_{\mathbf{w}}$ on $\mathbb{N}_0^m$ is defined as $\alpha \prec_{\mathbf{w}} \beta$ if and only if either $\mathrm{wd}(\alpha) < \mathrm{wd}(\beta)$ or $\mathrm{wd}(\alpha) = \mathrm{wd}(\beta)$ and $\alpha \prec_L \beta$, and similarly for the monomials.

Exercise 5.4. Show that wd is a degree function on $\mathbb{F}[X_1, \ldots, X_m]$ and that $\prec_{\mathbf{w}}$ is a reduction order that is isomorphic with $(\mathbb{N}, <)$.

Exercise 5.5. Let R be an $\mathbb{F}$-algebra with a weight function. Show that the set of elements of weight zero is equal to $\mathbb{F}^*$.

Example 5.6. Consider the $\mathbb{F}$-algebra

$$R = \mathbb{F}[X, Y]/(X^5 - Y^4 - Y).$$

Assume that R has a weight function ρ. Let x and y be the cosets in R of X and Y, respectively. Then $x^5 = y^4 + y$. Now $y \notin \mathbb{F}$, so $\rho(y) > 0$ by Exercise 5.5, and $\rho(y^4) = 4\rho(y) > \rho(y)$ by $(O.5)$. Thus $\rho(y^4 + y) = \rho(y^4)$ by $(O.2)$. Therefore

$$5\rho(x) = \rho(x^5) = \rho(y^4 + y) = 4\rho(y)$$

Thus the only possible solution is $\rho(x) = 4$ and $\rho(y) = 5$.

Exercise 5.7. Let $R = \mathbb{F}[X, Y]/(X^3Y + Y^3 + X)$. Show by the same reasoning as in the example above that $\rho(x) = 2$ and $\rho(y) = 3$ if there exists a weight function ρ on R. Prove that there exists no weight function on R.

Let $\mathcal{M}$ be the set of monomials in $X_1, \dots, X_m$. The *footprint* or *Δ-set* of a finite set $\mathcal{B}$ of polynomials is defined by

$$\Delta(\mathcal{B}) = \mathcal{M} \setminus \{\mathrm{lm}(BM) \mid B \in \mathcal{B}, B \neq 0, M \in \mathcal{M}\}.$$

(Here lm denotes the leading monomial.) If $\mathcal{B}$ is a Gröbner basis for the ideal I in R, then the cosets modulo I of the elements of the footprint $\Delta(\mathcal{B})$ form a basis of R/I.

Exercise 5.8. Let $\prec_{\mathbf{w}}$ be the lexicographical total weighted degree order on the monomials in X and Y with weights 4 and 5 for X and Y, respectively. Show that

$$\{X^i Y^j \mid i, j \in \mathbb{N}_0,\ j < 4\}$$

is the footprint of $X^5 + Y^4 + Y$ with respect to the reduction order $\prec_{\mathbf{w}}$.

Prove that the degree function wd is injective on this footprint. Let $(F_l \mid l \in \mathbb{N})$ be an enumeration of this footprint such that $\mathrm{wd}(F_l) < \mathrm{wd}(F_{l+1})$ for all l. Let $R = \mathbb{F}[X,Y]/(X^5 + Y^4 + Y)$. Let f_i be the coset of F_i in R. Thus $(f_l \mid l \in \mathbb{N})$ is a basis of R over $\mathbb{F}$. Define $\rho_l = 4i + 5j$ if $f_l = x^i y^j$.

Let $L(l)$ be the vector space with $f_1, \dots, f_l$ as basis. Let $l(i,j)$ be the smallest l such that $f_i f_j \in L(l)$. Prove that $\rho_l = \rho_i + \rho_j$ if $l = l(i,j)$.

Show that there exists a weight function on R as a conclusion of the above results or as a special case of the following.

Theorem 5.9. *Let I be an ideal in $\mathbb{F}[X_1, \dots, X_m]$ with Gröbner basis $\mathcal{B}$ with respect to $\prec_{\mathbf{w}}$. Suppose that the elements of the footprint of I have mutually distinct weighted degrees and that every element of $\mathcal{B}$ has two monomials of highest weighted degree in its support. Then there exists a weight function ρ on $R = \mathbb{F}[X_1, \dots, X_m]/I$ with the property that $\rho(f) = \mathrm{wd}(F)$, where f is the coset of F modulo I, for all polynomials F.*

Exercise 5.10. Let $R = \mathbb{F}[X,Y]/(X^a + Y^b + G(X,Y))$, where $\gcd(a,b) = 1$ and $\deg(G) < b < a$. Show that R has a weight function ρ such that $\rho(x) = b$ and $\rho(y) = a$.

Exercise 5.11. Let ρ be a weight function. Let $\Gamma = \{\rho(f) \mid f \in R,\ f \neq 0\}$. We may assume that the greatest common divisor of Γ is 1. Then Γ is called the set of *non-gaps*, and the complement of Γ in $\mathbb{N}_0$ is the set of *gaps*. Show that the number of gaps of the weight function of Example 5.10 is equal to $(a-1)(b-1)/2$.

5.2 A Bound on the Minimum Distance

We denote the coordinatewise multiplication on $\mathbb{F}_q^n$ by $*$. Thus $\mathbf{a} * \mathbf{b} = (a_1 b_1, \dots, a_n b_n)$ for $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{b} = (b_1, \dots, b_n)$. The vector space $\mathbb{F}_q^n$ becomes an $\mathbb{F}_q$-algebra with the multiplication $*$.

Let R be an *affine* $\mathbb{F}_q$-algebra, i.e., $R = \mathbb{F}_q[X_1, \dots, X_m]/I$, where I is an ideal of $\mathbb{F}_q[X_1, \dots, X_m]$. Let $\mathcal{P} = \{P_1, \dots, P_n\}$ consist of n distinct points of the zero set of I in $\mathbb{F}_q^m$. Consider the evaluation map

$$ev_{\mathcal{P}} : R \longrightarrow \mathbb{F}_q^n,$$

defined as $ev_{\mathcal{P}}(f) = (f(P_1), \dots, f(P_n))$.

Exercise 5.12. Show that $ev_{\mathcal{P}}$ is well defined and a morphism of $\mathbb{F}_q$-algebras, that means that this map is $\mathbb{F}_q$-linear and $ev_{\mathcal{P}}(fg) = ev_{\mathcal{P}}(f) * ev_{\mathcal{P}}(g)$ for all $f, g \in R$. Prove that the evaluation map is surjective.

Assume that R has an order function ρ. Then there exists a well-behaving sequence $(f_i \mid i \in \mathbb{N})$ of R over $\mathbb{F}_q$ by Exercise 5.2. So $\rho(f_i) < \rho(f_{i+1})$ for all i. Let $\mathbf{h}_i = ev_{\mathcal{P}}(f_i)$. Define

$$C(l) = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{c} \cdot \mathbf{h}_j = 0 \text{ for all } j \leq l\}.$$

The map $ev_{\mathcal{P}}$ is surjective, so there exists an N such that $C(l) = 0$ for all $l \geq N$.

Let $\mathbf{y} \in \mathbb{F}_q^n$. Consider

$$s_{ij}(\mathbf{y}) = \mathbf{y} \cdot (\mathbf{h}_i * \mathbf{h}_j).$$

Then $S(\mathbf{y}) = (s_{ij}(\mathbf{y}) \mid 1 \leq i, j \leq N)$ is the *matrix of syndromes* of $\mathbf{y}$.

Exercise 5.13. Prove that

$$S(\mathbf{y}) = HDH^\top,$$

where D is the $n \times n$ diagonal matrix with $\mathbf{y}$ on the diagonal and H is the $N \times n$ matrix with rows $\mathbf{h}_1, \dots, \mathbf{h}_N$. Use this fact to show that

$$\text{rank } S(\mathbf{y}) = \text{wt}(\mathbf{y}).$$

Let $L(l)$ be the vector space with basis $f_1, \dots, f_l$. Let $l(i,j)$ be the smallest l such that $f_i f_j \in L(l)$. Define

$$N(l) = \{(i,j) \mid l(i,j) = l + 1\}.$$

Let $\nu(l)$ be the number of elements of $N(l)$.

Exercise 5.14. Show that $i_1 < \dots < i_t \leq r$ and $j_t < \dots < j_1 \leq r$, if $(i_1, j_1), \dots, (i_t, j_t)$ is an enumeration of the elements of $N(l)$ in increasing order with respect to the lexicographical order.

Exercise 5.15. Suppose that $\mathbf{y} \in C(l) \setminus C(l+1)$. Prove that

$$s_{i_u j_v}(\mathbf{y}) = \begin{cases} 0 & \text{if } u + v \leq t \\ \text{not zero} & \text{if } u + v = t + 1. \end{cases}$$

Use this fact together with Exercises 5.13 and 5.14 to prove that

$$\text{wt}(\mathbf{y}) \geq \nu(l).$$

Define

$$d_{ORD}(l) = \min\{\nu(l') \mid l' \leq l\}$$

$$d_{ORD,\mathcal{P}}(l) = \min\{\nu(l') \mid l' \geq l,\ C(l') \neq C(l'+1)\}.$$

As a consequence of the definitions and Exercise 5.15 we get the following theorem.

Theorem 5.16. *The numbers $d_{ORD,\mathcal{P}}(l)$ and $d_{ORD}(l)$ are lower bounds for the minimum distance of $C(l)$:*

$$d(C(l)) \geq d_{ORD,\mathcal{P}}(l) \geq d_{ORD}(l).$$

Exercise 5.17. *Reed-Solomon codes.* Let $R = \mathbb{F}_q[X]$. Let ρ be the order function defined as $\rho(f) = \deg(f)$. Let α be a primitive element of $\mathbb{F}_q$. Let $n = q - 1$ and $\mathcal{P} = \{\alpha^0, \dots, \alpha^{n-1}\}$.

Prove that $(X^{i-1} \mid i \in \mathbb{N})$ is a well-behaving sequence and $l(i,j) = i+j-1$.

Show that $C(l)$ is a cyclic code with defining set $\{0, 1, \dots, l-1\}$ and $d_{ORD}(l) = l + 1$. Thus the BCH bound is obtained.

Exercise 5.18. Let ρ be a weight function and $(f_i \mid i \in \mathbb{N})$ a well-behaving sequence. Let $\rho_i = \rho(f_i)$. Show that $N(l) = \{(i,j) \mid \rho_i + \rho_j = \rho_{l+1}\}$.

Exercise 5.19. This is a continuation of Exercise 5.8 with $\mathbb{F} = \mathbb{F}_{16}$. Prove that $d_{ORD}(l) = \nu(l) = l - 5$ for all $l \geq 17$ and verify the numbers in the following table.

l	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ρ_l	0	4	5	8	9	10	12	13	14	15	16	17	18	19	20	21
$\nu(l)$	2	2	3	4	3	4	6	6	4	5	8	9	8	9	10	12
$d_{ORD}(l)$	2	2	3	3	3	4	4	4	4	5	8	8	8	9	10	12

Show that there are exactly 64 zeros of the ideal $X^5 + Y^4 + Y$ with coordinates in $\mathbb{F}_{16}$. Denote this zeroset by $\mathcal{P}$. Determine $d_{ORD,\mathcal{P}}(l)$ for all l.

Exercise 5.20. Suppose that ρ is a weight function. Let γ be the number of gaps. Show that $d_{ORD}(l) \geq l + 1 - \gamma$.

Exercise 5.21. *Reed-Muller codes.* Let $R = \mathbb{F}_q[X_1, \dots, X_m]$ and let ρ be the order function associated to the lexicographical total degree order on the monomials of R. Let $n = q^m$. Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be an enumeration of the q^m points of $\mathbb{F}_q^m$.

Show that $\nu(l) = \prod(\epsilon_i + 1)$ and $d_{ORD}(l) = (\sum \epsilon_i) + 1$ when $f_{l+1} = \prod X_i^{\epsilon_i}$.

Now suppose that $f_{l+1} = X_m^{r+1}$. Then $\{f_i | i \leq l\}$ is the set of monomials of degree at most r. The corresponding words $\{\mathbf{h}_i \mid i \leq l\}$ generate $RM_q(r, m)$, the Reed-Muller code over $\mathbb{F}_q$ of order r in m variables. So $C(l)$ is the dual of $RM_q(r, m)$ which is in fact equal to $RM_q((q-1)m - r - 1, r)$.

Write $r + 1 = \rho(q-1) + \mu$ with $\rho, \mu \in \mathbb{N}_0$ such that $\mu < q - 1$. Prove that $d(C(l)) = d_{ORD,\mathcal{P}}(l) = (\mu + 1)q^\rho$.

Notes

We use [9, 14] and Chapter 1 as a reference for the theory of Gröbner bases, and [31, 32] for the theory of error-correcting codes. The computer algebra packages Axiom [27], GAP [18] and Macaulay [36] are used for the computations.

The weight enumerator and MacWilliams identity is treated in [31, 32].

See the projects 6 on the Mathieu groups and 7 on Golay codes for more about automorphism groups of codes and its connection with designs.

For an algorithm to compute the automorphism group of a code we refer to [30].

For questions concerning complexity issues in coding theory we refer to [7]. The recent proof of the NP completeness of finding the minimum distance of a linear code is in [39]. This answers a problem posed in [11]. For cyclic codes there is an algorithm [8] to compute the weight enumerator that is much faster than the methods presented here.

For the tables of optimal q-ary codes for $q = 2$, 3 and 4, see [13]. There is an online connection to the latest state of the table [12] which can also be used to propose a new worldrecord. Brouwer's algorithm is incorporated in the coding theory package GUAVA [6, 35].

For finite geometry and projective systems we refer to [25, 38].

The treatment of the weight enumerator in Section 3.4 is from [28, 38]; this way of computing the weight distribution has been implemented by [10].

The treatment of the Mattson-Solomon polynomial can be found in [31, 32]. The proof of Proposition 4.7 is from [31, Chapter 6] or [32, §8.6]. The proof of Proposition 4.9 is from [32, §8.6 Theorem 24]. The relation with the ordinary Newton identities is explained in [32, Chap 8 §6 (52)].

The method in Section 4.2 to get the minimal codewords of cyclic codes is from [1, 2, 3, 4, 5]. This can be generalized to all linear codes as will be explained in the next chapter.

Goppa [19, 20, 21, 22, 23] used algebraic curves to construct codes. Nowadays, these codes are called geometric Goppa codes or algebraic geometry codes; they give asymptotically good codes, even better than the Gilbert-Varshamov bound [38]. The mathematics is quite deep and abstract. For the construction and the parameters of these codes one needs the theory of algebraic curves or algebraic function fields of one variable [37], in particular, the *Riemann-Roch Theorem.* The asymptotically good codes require the knowledge of *modular curves.* Several authors [15, 16, 17, 26, 29] have proposed a more elementary approach to algebraic geometry codes and this new method has much to do with Gröbner bases [34].

The notion of order and weight functions and its relation with coding theory is developed in [24, 34].

Section 5 is from [26, 29, 34]. Theorem 5.9 is from [34]. The values of an order function form a semigroup in the case of a weight function. The order bound is called the Feng-Rao bound and is computed in terms of the properties of the semigroup [29]. The way Reed-Muller codes are treated in Exercise 5.21 is from [24, 33].

A classical treatment of algebraic geometry codes is given in [37, 38].

References

1. D. Augot (1996): ***Description of minimum weight codewords of cyclic codes by algebraic systems***, **Finite Fields and their Appl. 2, 138–152.**
2. D. Augot (1994): ***Algebraic characterization of minimum codewords of cyclic codes***, **pp. 46 in Proc. IEEE ISIT'94, Trondheim, Norway, June 1994.**

3. D. Augot (1995): *Newton's identities for minimum codewords of a family of alternant codes*, preprint.
4. D. Augot, P. Charpin, and N. Sendrier (1990): *Weights of some binary cyclic codewords throughout Newton's identities*, pp. 65–75 in Eurocode '90, Lecture Notes Comp. Sc. **514**, Springer-Verlag, Berlin Heidelberg New York.
5. D. Augot, P. Charpin, and N. Sendrier (1992): *Studying the locator polynomial of minimum weight codewords of BCH codes*, IEEE Trans. Inform. Theory **38**, 960–973.
6. R. Baart, J. Cramwinckel, and E. Roijackers (1994): *GUAVA, a coding theory package*, Delft Univ. Technology.
7. A. Barg: *Complexity issues in coding theory*, to appear in Handbook of Coding Theory, (V.S. Pless, W.C. Huffman and R.A. Brualdi eds.), Elsevier.
8. A. Barg and I. Dumer (1992): *On computing the weight spectrum of cyclic codes*, IEEE Trans. Inform. Theory **38**, 1382–1386.
9. T. Becker and V. Weispfenning (1993): *Gröbner Bases; a Computational Approach to Commutative Algebra*, Springer-Verlag, Berlin Heidelberg New York.
10. M. Becker and J. Cramwinckel (1995): *Implementation of an algorithm for the weight distribution of block codes*, Modelleringcolloquium, Eindhoven Univ. Technology.
11. E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg (19978): *On the inherent intractibility of certain coding problems*, IEEE Trans. Inform. Theory **24**, 384–386.
12. A.E. Brouwer, http://www.win.tue.nl/win/math.dw.voorlincod.html
13. A.E. Brouwer and T. Verhoeff (1993): *An updated table of minimum-distance bounds for binary linear codes*, IEEE Trans. Inform. Theory **39**, 662–677.
14. D. Cox, J. Little, and D. O'Shea (1992): *Ideals, Varieties and Algorithms; An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer-Verlag, Berlin Heidelberg New York.
15. G.-L. Feng and T.R.N. Rao (1994): *A simple approach for construction of algebraic-geometric codes from affine plane curves*, IEEE Trans. Inform. Theory **40**, 1003–1012.
16. G.-L. Feng and T.R.N. Rao (1995): *Improved geometric Goppa codes Part I, Basic Theory*, IEEE Trans. Inform. Theory **41**, 1678–1693.
17. G.-L. Feng, V. Wei, T.R.N. Rao, and K.K. Tzeng (1994): *Simplified understanding and efficient decoding of a class of algebraic-geometric codes*, IEEE Trans. Inform. Theory **40**, 981–1002.
18. M. Schönert et al. (1994): GAP – *Groups, Algorithms and Programming*, version 3, release 4, Lehrstuhl D für Mathematik, RWTH Aachen.
19. V.D. Goppa (1977): *Codes associated with divisors*, Probl. Peredachi Inform. **13** (1) 33–39. Translation: Probl. Inform. Transmission **13**, 22–26.
20. V.D. Goppa (1981): *Codes on algebraic curves*, Dokl. Akad. Nauk SSSR **259**, 1289–1290. Translation: Soviet Math. Dokl. **24**, 170–172.
21. V.D. Goppa (1982): *Algebraico-geometric codes*, Izv. Akad. Nauk SSSR **46**. Translation: Math. USSR Izvestija **21**, 75–91, 1983.
22. V.D. Goppa (1984): *Codes and information*, Usp. Mat. Nauk **39**, No. 1, 77–120. Translation: Russian Math. Surveys **39**, 87–141, 1984.
23. V.D. Goppa (1991): *Geometry and codes*, Mathematics and its Applications **24**, Kluwer Acad. Publ., Dordrecht.
24. P. Heijnen and R. Pellikaan (1998): *Genenerallized Hamming weights of q-ary Reed-Muller codes*, IEEE Trans. Inform. Theory **44**, 181–196.
25. J.W.P. Hirschfeld and J.A. Thas (1991): *General Galois Geometries*, Oxford University Press, Oxford.

26. T. Høholdt, J. H. van Lint, and R. Pellikaan: *Algebraic geometry codes*, to appear in Handbook of Coding Theory, (V.S. Pless, W.C. Huffman and R.A. Brualdi eds.), Elsevier.
27. R. D. Jenks and R. S. Sutor (1992): *Axiom. The Scientific Computation System*, Springer-Verlag, New York Berlin Heidelberg.
28. G. L. Katsman and M. A. Tsfasman (1987): *Spectra of algebraic-geometric codes*, Probl. Peredachi Inform **23** (4) 19–34. Translation: Probl. Inform. Transmission **23**, 262–275.
29. C. Kirfel and R. Pellikaan (1995): *The minimum distance of codes in an array coming from telescopic semigroups*, IEEE Trans. Inform. Theory **41**, 1720–1732.
30. J. Leon (1982): *Computing the automorphism groups of error-correcting codes*, IEEE Trans. Inform. Theory **28**, 496–511.
31. J. H. van Lint (1982): *Introduction to Coding Theory*, Graduate Texts in Math. **86**, Springer-Verlag, Berlin Heidelberg New York.
32. F. J. MacWilliams and N. J. A. Sloane (1977): *The Theory of Error-Correcting Codes*, North-Holland Math. Library **16**, North-Holland, Amsterdam.
33. R. Pellikaan (1996): *The shift bound for cyclic, Reed-Muller and geometric Goppa codes*, pp. 155–175 in Proceedings AGCT-4, Luminy 1993, de Gruyter, Berlin.
34. R. Pellikaan (1996): *On the existence of order functions*, submitted to the proceedings of the Second Shanghai Conference on Designs, Codes and Finte Geometry.
35. J. Simonis (1994): *GUAVA: A computer algebra package for coding theory*, pp. 165–166 in Proc Fourth Int. Workshop Algebraic Combinatorial Coding Theory, Novgorod, Russia, Sept. 11–17, 1994.
36. Ma. Stillman, Mi. Stillman, and D. Bayer, *Macaulay User Manual.*
37. H. Stichtenoth (1993): *Algebraic Function Fields and Codes*, Universitext, Springer-Verlag, Berlin Heidelberg New York.
38. M. A. Tsfasman and S.G. Vlăduţ, (1991): *Algebraic-geometric codes*, Mathematics and its Application **58**, Kluwer Acad. Publ., Dordrecht.
39. A. Vardy (1997): *The intractibility of computing the minimum distance of a code*, IEEE Trans. Inform. Theory **43**, 1757–1766.

Chapter 11. Gröbner Bases for Decoding

Mario de Boer and Ruud Pellikaan

1. Introduction

From the previous chapter one might get the impression that the theory of error-correcting codes is equivalent to the theory of finite geometry or arrangements over finite fields. This is not true from a practical point of view. A code is useless without a decoding algorithm. For engineers the total performance of the encoding and decoding scheme is important.

An introduction to the decoding problem is given in Section 2. In Section 3 we first restrict ourselves to cyclic codes where the system of syndrome equations can be explicitly solved using Gröbner basis techniques and later, in Section 5, to arbitrary linear codes. Although this method decodes up to half the true minimum distance, the complexity is not polynomial, because there is no polynomial algorithm known to compute Gröbner bases. The algorithms of Euclid, Sugiyama, and Berlekamp-Massey give an efficient way to decode cyclic codes by solving the key equation.

All references and suggestions for further reading will again be given in the notes at the end of this Chapter.

2. Decoding

Let C be a linear code. Decoding is the inverse operation of encoding. A *decoder* is a map

$$\mathcal{D} : \mathbb{F}_q^n \longrightarrow C \cup \{?\},$$

such that $\mathcal{D}(\mathbf{c}) = \mathbf{c}$ for all $\mathbf{c} \in C$. Let $\mathbf{y}$ be a *received word.* Then $\mathcal{D}(\mathbf{y})$ is a codeword or equal to ?, in case of a *decoding failure*

Decoding by *error detection* does the following. Let H be a parity check matrix of C. The output of the decoder is $\mathbf{y}$ if $\mathbf{y}H^\top = 0$, and ? otherwise.

If the received word $\mathbf{y}$ is again a codeword, but not equal to the one sent, then the decoder gives $\mathbf{y}$ as output and we have a *miscorrection* also called a *decoding error.*

Let $C \subseteq \mathbb{F}_q^n$ be the code with minimum distance d that is used to transmit information over a noisy channel. If the codeword $\mathbf{c}$ is transmitted at one side of the channel and $\mathbf{y}$ is received at the other end, then we say that the *error* $\mathbf{e} = \mathbf{y} - \mathbf{c}$ has occurred:

$$\mathbf{y} = \mathbf{c} + \mathbf{e}.$$

A decoder $\mathcal{D}$ is called a *minimum distance decoder* if $\mathcal{D}(\mathbf{y})$ is a codeword that is nearest to $\mathbf{y}$ with respect to the Hamming metric for all $\mathbf{y}$.

Minimum distance decoding is similar to finding a codeword of minimal weight. If $\mathbf{y}$ is a received word, then one has to find a word in the coset $\mathbf{y}+C$ of minimal weight. Such a word is called a *coset leader*. To store a list of all coset leaders requires a memory of q^{n-k} such elements and is only efficient for codes of small redundancy.

If the Hamming weight of the error-vector is at most $\lfloor (d-1)/2 \rfloor$, then $\mathbf{c}$ is the unique codeword which has the smallest distance to $\mathbf{y}$, so the error can be corrected. The value $t = \lfloor (d-1)/2 \rfloor$ is called the *error-correcting capability* or *capacity* of the code.

Let H be a parity check matrix for C, so $\mathbf{c}H^{\top} = 0$ for all $\mathbf{c} \in C$. After receiving $\mathbf{y}$ one computes the vector of *syndromes*

$$\mathbf{s} = \mathbf{y}H^{\top}.$$

Since $\mathbf{y} = \mathbf{c} + \mathbf{e}$ we have that $\mathbf{s} = \mathbf{y}H^{\top} = \mathbf{c}H^{\top} + \mathbf{e}H^{\top} = \mathbf{e}H^{\top}$ and the problem becomes: given $\mathbf{s}$, find a vector $\mathbf{e}$ of lowest Hamming weight such that $\mathbf{e}H^{T} = \mathbf{s}$.

A decoder $\mathcal{D}$ is called a *bounded distance decoder* that *corrects t errors* if $\mathcal{D}(\mathbf{y})$ is a codeword that is nearest to $\mathbf{y}$ for all $\mathbf{y}$ such that $d(\mathbf{y}, C) \leq t$. We say that $\mathcal{D}$ *decodes up to half the minimum distance* if it corrects $\lfloor (d-1)/2 \rfloor$ errors.

Proposition 2.1. *Let C be a linear code in $\mathbb{F}_q^n$ with parity check matrix H. Suppose we have a received word $\mathbf{y}$ with error-vector $\mathbf{e}$ and we know a set J with at most $d(C)-1$ elements and that contains the set of error positions. Then the error-vector $\mathbf{e}$ is the unique solution for $\mathbf{x}$ of the following linear equations:*

$$\mathbf{x}H^{\top} = \mathbf{y}H^{\top} \quad \textit{and} \quad x_j = 0 \ \textit{for} \ j \notin J.$$

Exercise 2.2. Prove Proposition 2.1 and deduce that the syndrome of a received word with at most $\lfloor (d-1)/2 \rfloor$ errors is unique.

Proposition 2.1 shows that error decoding can be reduced to the problem of finding the error positions. If we want to decode all received words with t errors, then there are $\binom{n}{t}$ possible t-sets of error positions one has to consider. This number grows exponentially with n if t/n tends to a nonzero real number. The decoding problem is hard. Only for special families of codes this problem has an efficient solution with practical applications. We will consider only bounded distance decoders.

Exercise 2.3. Assume that the channel is a *q-ary symmetric channel*. This means that the probability that the symbol $x \in \mathbb{F}_q$ is changed in the symbol $y \in \mathbb{F}_q$ is the same for all $x, y \in \mathbb{F}_q$ and $x \neq y$, and does not depend on the position. The probability that a fixed symbol is changed in another symbol, distinct from the original one, is called the *crossover* probability and is denoted by P. Prove that the probability that an error vector $\mathbf{e}$ is equal to the word $\mathbf{c}$ of weight t is given by

$$\mathrm{Prob}\{\mathbf{e} = \mathbf{c}\} = \left(\frac{P}{q-1}\right)^t (1-P)^{n-t}.$$

Show that the *undetected error probability* is given by

$$W_C\left(1-P, \frac{P}{q-1}\right) - (1-P)^n,$$

where $W_C(X, Y)$ is the homogeneous weight enumerator of C.

3. Decoding Cyclic Codes with Gröbner Bases

Let C be an $[n, k, d]$ cyclic code with generator polynomial $g(X)$ and defining set $J = \{j_1, \ldots, j_r\}$. Let $\mathbb{F}_{q^e}$ be an extension of $\mathbb{F}_q$ that contains all the zeros of $g(X)$. Let $\alpha \in \mathbb{F}_{q^e}$ be a primitive n-th root of unity. Then a parity check matrix of C is

$$H = \begin{pmatrix} 1 & \alpha^{j_1} & \alpha^{2j_1} & \cdots & \alpha^{(n-1)j_1} \\ 1 & \alpha^{j_2} & \alpha^{2j_2} & \cdots & \alpha^{(n-1)j_2} \\ 1 & \alpha^{j_3} & \alpha^{2j_3} & \cdots & \alpha^{(n-1)j_3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{j_r} & \alpha^{2j_r} & \cdots & \alpha^{(n-1)j_r} \end{pmatrix}.$$

Now let $\mathbf{e} = e(X)$ be an error-vector of a received word $\mathbf{y} = y(X)$. Then $\mathbf{s} = \mathbf{y}H^\top = \mathbf{e}H^\top$ and

$$s_i = y(\alpha^{j_i}) = e(\alpha^{j_i})$$

is the i-th component of $\mathbf{s}$ for $i = 1, \ldots, r$. It is more convenient to consider the extension $\hat{H}$ of the matrix H, where $\hat{H}$ is the $n \times n$ matrix with i-th row

$$(1\ \alpha^i\ \alpha^{2i}\ \cdots\ \alpha^{(n-1)i})$$

for $i = 1, \ldots, n$. Define $\hat{\mathbf{s}} = \mathbf{e}\hat{H}^\top$. The j-th component of $\hat{\mathbf{s}}$ is

$$\hat{s}_j = e(\alpha^j) = \sum_{i=0}^{n-1} \alpha^{ij}$$

for $j = 1, \ldots, n$. If $j \in J(C)$, then $\hat{s}_j = e(\alpha^j) = y(\alpha^j)$, so these syndromes are *known*.

From now on $\hat{s}_j$ will be denoted by s_j. Notice that the old s_i is now denoted by s_{j_i}.

Let $\mathbf{e} = e(X)$ be an error-vector with error positions $i_1, i_2, \ldots, i_t$ and error values $e_{i_1}, e_{i_2}, \ldots, e_{i_t}$. Then the known syndromes will be

$$s_j = \sum_{m=1}^{t} e_{i_m} (\alpha^{i_m})^j, \qquad j \in J(C).$$

Consider the following system of equations over $\mathbb{F}_{q^e}[X_1, \ldots, X_v, Y_1, \ldots, Y_v]$:

$$\mathcal{S}(\mathbf{s}, v) = \begin{cases} \sum_{m=1}^{v} Y_m X_m^j & = & s_j & \text{for } j \in J \\ Y_m^q & = & Y_m & \text{for } m = 1, \ldots, v \\ X_m^n & = & 1 & \text{for } m = 1, \ldots, v. \end{cases}$$

Conclude that $X_m = \alpha^{i_m}$ and $Y_m = e_{i_m}$ for $m = 1, \ldots, t$ is a solution of $\mathcal{S}(\mathbf{s}, t)$.

Exercise 3.1. Show that the equation $\sum_{m=1}^{v} Y_m X_m^{jq} = s_{jq}$ is a consequence of $\mathcal{S}(\mathbf{s}, v)$ for all $j \in J$.

Example 3.2. Let $J = \{1, 2\}$. If C is a cyclic code with defining set J, then its minimum distance is at least 3 by the BCH bound. So one can correct at least 1 error. The equations

$$\begin{cases} Y_1 X_1 & = & s_1 \\ Y_1 X_1^2 & = & s_2 \end{cases}$$

imply that the error position is $x_1 = s_2/s_1$ if there is exactly one error. If moreover $q = 2$, then $s_2 = s_1^2$, so $x_1 = s_1$.

We have the following.

Proposition 3.3. *Suppose that t errors occurred and $t \leq (d-1)/2$. Then the system $\mathcal{S}(\mathbf{s}, v)$ over $\mathbb{F}_{q^e}$ has no solution when $v < t$, and a unique solution, up to permutation, corresponding to the error-vector of lowest weight that satisfies the syndrome equations when $v = t$. The X_i of the solution are the error-locators and the Y_i the corresponding error values. If $v > t$, then for every j the system has a solution with $X_1 = \alpha^j$.*

Exercise 3.4. Prove Proposition 3.3 using Proposition 2.1.

The system $\mathcal{S}(\mathbf{s}, v)$ defines an ideal in the ring $\mathbb{F}_{q^e}[X_1, \ldots, X_v, Y_1, \ldots, Y_v]$. By abuse of notation we denote this ideal also by $\mathcal{S}(\mathbf{s}, v)$. The zeroset of this ideal gives the error-vector that occurred during the transmission. Gröbner basis techniques can be used to find the solutions of the equations.

Let $\prec_L$ be the lexicographic order with $Z_1 \prec_L Z_2 \prec_L \cdots \prec_L Z_w$. Then $\prec_L$ is an ***elimination order***, that is to say it satisfies the following property.

Proposition 3.5. ***Let I be an ideal in $\mathbb{F}[Z_1, Z_2, \ldots, Z_w]$. Let $\mathcal{G}$ be a Gröbner basis of I with respect to $\prec_L$. Then $\mathcal{G} \cap \mathbb{F}[Z_1, Z_2, \ldots, Z_i]$ is a Gröbner basis of $I \cap \mathbb{F}[Z_1, Z_2, \ldots, Z_i]$.***

Let I be an ideal in $\mathbb{F}[Z_1, Z_2, \ldots, Z_w]$ with finitely many zeros over $\bar{\mathbb{F}}$ which are all defined over $\mathbb{F}$. Let V be the zeroset in $\mathbb{F}^w$ of the ideal I. Then the zeroset of $I \cap \mathbb{F}[Z_1, Z_2, \ldots, Z_i]$ is equal to the projection of V on the first i coordinates. This fact and Proposition 3.5 have a direct application to our problem of finding the solutions to system $\mathcal{S}(\mathbf{s}, v)$. Indeed, if $(x_1, \ldots, x_v)$ is the X-part of a solution to $\mathcal{S}(\mathbf{s}, v)$, then also any permutation of the x_i will be a solution (apply the same permutation to the Y-part of the solution). Hence every error-locator will appear as the first coordinate of a solution to $\mathcal{S}(\mathrm{s}, v)$. Thus we have sketched the proof of the following.

Proposition 3.6. *Suppose that t errors occurred and $t \leq (d-1)/2$. Let $g(X_1)$ be the monic generator of the ideal $\mathcal{S}(\mathbf{s}, t) \cap \mathbb{F}_{q^e}[X_1]$. Then the zeros of g are the error-locators.*

Before giving the final algorithm for the decoding, we must worry about one more thing: we assumed we knew how many errors occurred (the v occurring in system $\mathcal{S}(\mathbf{s}, v)$). Now note that the work required to solve the system $\mathcal{S}(\mathbf{s}, v)$ for large v is much more than for small v, and remark that in general words with many errors occur less often than words with few or no errors. The following theorem leads the way to an algorithm that implements this idea.

Theorem 3.7. *Suppose t errors occurred and $t \leq (d-1)/2$. Denote the monic error-locator polynomial by $l(X_1)$, that is, $l(x) = 0$ if and only if x is an error-locator. Let $g(X_1)$ be the monic generator of the ideal $\mathcal{S}(\mathbf{s}, v) \cap \mathbb{F}_{q^e}[X_1]$, with $\mathcal{S}(\mathbf{s}, v)$ the ideal in $\mathbb{F}_{q^e}[X_1, \ldots, X_v, Y_1, \ldots, Y_v]$. Then*

$$g(X_1) = \begin{cases} 1 & \text{if } v < t \\ l(X_1) & \text{if } v = t \\ X_1^n - 1 & \text{if } v > t \end{cases}$$

Exercise 3.8. Show that in Proposition 3.3 and Theorem 3.7 it is allowed to replace the assumption '$t \leq (d-1)/2$' by the weaker statement 'the received word has a unique closest codeword'.

Exercise 3.9. Let $\mathcal{S}'(\mathbf{s}, v)$ be the system of equations which is obtained by replacing the equation $Y_m^q = Y_m$ in $\mathcal{S}(\mathbf{s}, v)$ by $Y_m^{q-1} = 1$ for all $m = 1, \ldots, v$. So the variables Y_m disappear if $q = 2$. How should Proposition 3.3 and Theorem 3.7 be restated for $\mathcal{S}'(\mathbf{s}, v)$?

We are now ready to state the algorithm to decode cyclic codes.

Algorithm 3.10.

```
input($\mathbf{y}$);
$\mathbf{s} := \mathbf{y}H^{\top}$;
if $s_j = 0$ for all $j \in J$
then output($\mathbf{y}$); stop; {no errors occurred}
else $v := 1$;
```

```
$\mathcal{G} := \{1\}$;
while $1 \in \mathcal{G}$ do
  $\mathcal{S} := \{\sum_{m=1}^{v} Y_m X_m^j - s_j, j \in J\} \cup \{Y_m^q - Y_m, X_m^n - 1,\ m = 1,\dots,v\}$;
  $\mathcal{G} :=$ Gröbner($\mathcal{S}$);
  $v := v + 1$;
od;
{$1 \notin \mathcal{G}$ so there are solutions}
$g(X_1) :=$ the unique element of $\mathcal{G} \cap \mathbb{F}_{q^e}[X_1]$};
if deg($g(X_1)$) $> v$
then output(?); stop { too many errors }
else error-locators := {zeros of $g(Z_1)$}
     find error-vector $\mathbf{e}$ by solving the linear equations
     as in Proposition 2.1
     output($\mathbf{y} - \mathbf{e}$)
```

We will treat an example in the project on the Golay codes.

3.1 One-Step Decoding of Cyclic Codes

In the system of equations $\mathcal{S}(\mathbf{s}, v)$ the syndromes s_j are considered to be known constants. In this section we treat the syndromes as variables and consider the corresponding system of equations

$$\mathcal{S}(v) = \left\{ \begin{array}{rcll} \sum_{m=1}^{v} Y_m X_m^j & = & S_j & \text{for } j \in J \\ Y_m^q & = & Y_m & \text{for } m = 1,\dots,v \\ X_m^n & = & 1 & \text{for } m = 1,\dots,v. \end{array} \right.$$

to define an ideal in the ring

$$\mathbb{F}_{q^e}[X_1,\dots,X_v,Y_1,\dots,Y_v,S_j,j \in J].$$

Of course, this has the advantage that we have to solve these equations only once, and that this can be done before we start to use the code. This is called the *preprocessing* of the decoding algorithm. In the actual running of the algorithm the values of the syndromes s_j of a received word are substituted in the variables S_j for $j \in J$.

Exercise 3.11. Let $\prec$ be a reduction order on the monomials $X_1,\dots,X_v$, $Y_1,\dots,Y_v$ and $S_j, j \in J$ such that the variables $S_j, j \in J$ are larger than $X_1,\dots,X_v$ and $Y_1,\dots,Y_v$. Show that $\mathcal{S}(v)$ is a Gröbner basis with respect to $\prec$.

The exercise gives the impression that we are done. But we have to eliminate the variables $X_2,\dots,X_v$ and $Y_1,\dots,Y_v$. Therefore the variables X_1, $S_j, j \in J$ need to be smaller than $X_2,\dots,X_v$, $Y_1,\dots,Y_v$.

As an example, we have applied one-step decoding to binary cyclic codes with defining sets $\{1,3\}$, $\{1,3,5\}$ and $\{1,3,5,7\}$, respectively. Remark that the complete defining sets contain $\{1,2,3,4\}$, $\{1,2,3,4,5,6\}$ and $\{1,\dots,8\}$, respectively. From the BCH-bound we know that these codes can correct 2, 3 and 4 errors, respectively. The Gröbner basis is computed with a lexicographic order in a way such that the basis contains a polynomial in X_1 and the syndrome-variables S_j. We consider binary codes. Thus the error values are always 1. Therefore we delete the variables Y_i in the equations. The equations of the form $X_m^n = 1$ are also left out. So the number of solutions is not finite anymore. The results are as follows.

Example 3.12. $q = 2$, $\{1,3\} \subseteq J(C)$.

$$\mathcal{S} = \left\{ \begin{array}{ccccccc} X_1 & + & X_2 & - & S_1 & = & 0 \\ X_1^3 & + & X_2^3 & - & S_3 & = & 0 \end{array} \right.$$

Order: $X_2 > X_1 > S_3 > S_1$
Error-locator polynomial with $X = X_1$:

$$S_1X^2 + S_1^2X + (S_1^3 + S_3).$$

Example 3.13. $q = 2$, $\{1,3,5\} \subseteq J(C)$.

$$\mathcal{S} = \left\{ \begin{array}{ccccccccc} X_1 & + & X_2 & + & X_3 & - & S_1 & = & 0 \\ X_1^3 & + & X_2^3 & + & X_3^3 & - & S_3 & = & 0 \\ X_1^5 & + & X_2^5 & + & X_3^5 & - & S_5 & = & 0 \end{array} \right.$$

Order: $X_3 > X_2 > X_1 > S_5 > S_3 > S_1$
Error-locator polynomial:

$$(S_3 + S_1^3)X^3 + (S_3S_1 + S_1^4)X^2 + (S_5 + S_3S_1^2)X + (S_5S_1 + S_3^2 + S_3S_1^3 + S_1^6).$$

Example 3.14. $q = 2$, $\{1,3,5,7\} \subseteq J(C)$.

$$\mathcal{S} = \left\{ \begin{array}{ccccccccccc} X_1 & + & X_2 & + & X_3 & + & X_4 & - & S_1 & = & 0 \\ X_1^3 & + & X_2^3 & + & X_3^3 & + & X_4^3 & - & S_3 & = & 0 \\ X_1^5 & + & X_2^5 & + & X_3^5 & + & X_4^5 & - & S_5 & = & 0 \\ X_1^7 & + & X_2^7 & + & X_3^7 & + & X_4^7 & - & S_7 & = & 0 \end{array} \right.$$

Order: $X_4 > X_3 > X_2 > X_1 > S_7 > S_5 > S_3 > S_1$
Error-locator polynomial:

$$\begin{gathered} (S_1^6 + S_3^2 + S_5S_1 + S_3S_1^3)X^4 + (S_5S_1^2 + S_3^2S_1 + S_3S_1^4 + S_1^7)X^3 + \\ (S_7S_1 + S_5S_3 + S_3S_1^5 + S_1^8)X^2 + (S_7S_1^2 + S_5S_1^4 + S_3^3 + S_3S_1^6)X + \\ (S_7S_3 + S_7S_1^3 + S_5^2 + S_5S_3S_1^2 + S_5S_1^5 + S_3^3S_1 + S_3S_1^7 + S_1^{10}). \end{gathered}$$

Example 3.15. The error-locator polynomial for the 6-error correcting binary BCH code took four hours using Axiom. The coefficient of X^i has 20, 20, 22, 22, 20, 24 and 46 terms for $i = 6, 5, \dots, 1$ and 0, respectively.

Exercise 3.16. Give S_i weighted degree i and let $\mathrm{wd}(X) = 1$. Notice that in the above examples the error-locator polynomial is homogeneous of total weighted degree $\binom{t+1}{2}$ if the BCH bound is $2t+1$. Show that this is always the case.

Looking at the formulas for the 2, 3 and 4 error-correcting BCH codes one gets the impression that the number of terms grows exponentially (we do not know whether this is a fact). Thus specializing the values for the syndromes still would not give a decoding algorithm of polynomial complexity.

It is a priori not clear that substituting values for the syndromes in the variables after elimination gives the same answer as the original method with the syndromes as constants.

To make this point clear we introduce some notation. Let $\mathcal{G}$ be a subset of the polynomial ring in the variables $S_j, j \in J$, $X_1, \ldots, X_v$ and more. Then $\mathcal{G}_1$ is the subset of $\mathcal{G}$ of polynomials in the variables $S_j, j \in J$ and X_1 only. Let $\mathbf{s} = (s_j, j \in J)$ be a vector with coordinates in $\mathbb{F}_q$. Then $\mathcal{G}_1(\mathbf{s})$ is the set obtained from $\mathcal{G}_1$ by substituting the value s_j in S_j for all elements of $\mathcal{G}_1$ and $j \in J$.

Let $\prec_E$ be an elimination order on the monomials $X_1, \ldots, X_v$, $Y_1, \ldots, Y_v$ and $S_j, j \in J$ with the variables $X_1, \ldots, X_v$ and $Y_1, \ldots, Y_v$ larger than $S_j, j \in J$. That the one-step method works is stated as a fact in the following

Theorem 3.17. *Let $\mathcal{G}$ be a Gröbner basis of $\mathcal{S}(t)$ with respect to $\prec_E$. Let* $\mathbf{y}$ *be a received word such that t errors occurred. Let* s *be its syndrome. Assume that the closest codeword to* $\mathbf{y}$ *is unique. Then $\mathcal{G}_1$ is the Gröbner basis of $\mathcal{S}(t) \cap \mathbb{F}_q[X_1, S_j, j \in J]$ and $\mathcal{G}_1(\mathrm{s})$ is a (nonreduced) Gröbner basis and the error-locator polynomial is an element of $\mathcal{G}_1(\mathrm{s})$.*

The proof relies on the fact that $\mathcal{S}(t)$ has a finite number of solutions.

4. The Key Equation

Let C be a cyclic code of length n such that $\{1, 2, \ldots, \delta - 1\} \subset J(C)$. From the BCH bound we see that the minimum distance of C is at least δ. In this section we will give a decoding algorithm for such a code, which has an efficient implementation and is used in practice. A drawback of the algorithm is that it only corrects errors of weight at most $(\delta - 1)/2$, whereas the true minimum distance can be larger than δ. An example of this phenomenon will be treated in the project on the Golay codes.

The algorithms in this section work for cyclic codes that have any $\delta - 1$ consecutive elements in their complete defining set. We leave it to the reader to make the necessary adjustments in the case where these elements are not $\{1, 2, \ldots, \delta - 1\}$.

Let α be a primitive n-th root of unity. Let $\mathbf{c} = c(X) \in C$ be the transmitted codeword that is received as $\mathbf{y} = y(X) = c(X) + e(X)$, with

$w = wt(\mathbf{e}) \leq (\delta - 1)/2$. The support of $\mathbf{e}$ will be denoted by I. We then can compute the syndromes

$$s_i = A_i = e(\alpha^i) = y(\alpha^i) \qquad \text{for } i \in J(C),$$

where the A_i are the coefficients of the MS polynomial of $e(X)$, see Section 4.1. Since $\{1, 2, \ldots, \delta-1\} \subseteq J(C)$ and $2w \leq \delta-1$ we know all $A_1, A_2, \ldots, A_{2w}$. Write σ_i for the i-th symmetric function of the error positions and form the following set of generalized Newton identities, see Proposition 4.9, Chapter 10:

$$\left\{\begin{array}{llllllllll} A_{v+1} & + & \sigma_1 A_v & + & \cdots & + & \sigma_v A_1 & = & 0 \\ A_{v+2} & + & \sigma_1 A_{v+1} & + & \cdots & + & \sigma_v A_2 & = & 0 \\ & & & & \vdots & & & \vdots & \vdots \\ A_{2v} & + & \sigma_1 A_{2v-1} & + & \cdots & + & \sigma_v A_v & = & 0. \end{array}\right. \tag{4.1}$$

From the system with $v = w$ we have to find the σ_i. After we have done this, we can find the polynomial

$$\sigma(Z) = 1 + \sigma_1 Z + \sigma_2 Z^2 + \cdots + \sigma_w Z^w,$$

which has as its zeros the reciprocals of the error locations. Finding the zeros of this polynomial is an easy task. We return to the problem of finding the coefficients σ_i.

Exercise 4.1. Consider the system of equations (4.1) as linear in the unknown $\sigma_1, \ldots, \sigma_w$ with coefficients in $\mathbb{F}_q(A_1, \ldots, A_w)$, the field of rational functions in $A_1, \ldots, A_w$, which are treated now as variables. Then

$$\sigma_i = \frac{\Delta_i}{\Delta_0},$$

where Δ_i is the determinant of a certain $w \times w$ matrix according to Cramer's rule. Then the Δ_i are polynomials in the A_i. Conclude that

$$\Delta_0 X^w + \Delta_1 X^{w-1} + \cdots + \Delta_w$$

is a closed form of the *generic* error-locator polynomial.

Substitute $A_{2i+1} = S_{2i+1}$ and $A_{2i} = S_i^2$ and compare the result with Examples 3.12, 3.13 and 3.14.

Exercise 4.2. Show that the matrix $(A_{i+j-1} | 1 \leq i, j \leq v)$ is nonsingular if and only if $v = w$, the number of errors. Hint: Try to write the matrix as a triple product of matrices of known rank as done in Exercise 5.13.

The algorithm of *Arimoto-Peterson-Gorenstein-Zierler* (APGZ) solves the systems of linear equations (4.1) for $v = 1, \ldots, w$ by Gaussian elimination.

Exercise 4.3. What is the complexity of the algorithm of APGZ?

Write

$$S(Z) = \sum_{i=1}^{\delta-1} A_i Z^{i-1},$$

then an alternative way of formulating (4.1) is that there exist polynomials $q(Z)$ and $r(Z)$ such that

$$\sigma(Z)S(Z) = q(Z)Z^{\delta-1} + r(Z), \quad \deg(r(Z)) \le w-1,$$

or that there exists a polynomial $\omega(Z)$ of degree at most $w-1$ such that

$$\omega(Z) \equiv \sigma(Z)S(Z) \bmod Z^{\delta-1}. \tag{4.2}$$

This is called the *key equation.*

Exercise 4.4. Check that

$$\omega(Z) = \sum_{i\in I} e_i\alpha^i \prod_{j\in I\setminus\{i\}} (1-\alpha^j Z),$$

by rewriting $\omega(Z)/\sigma(Z) \bmod Z^{\delta-1}$.

Exercise 4.5. Let $\sigma'(Z)$ be the formal derivative of $\sigma(Z)$. Show *Forney's formula* for the error values:

$$e_i = -\frac{\omega(\alpha^{-i})}{\sigma'(\alpha^{-i})}$$

for all error positions i. The polynomial $\omega(Z)$ is called the *error evaluator polynomial.*

We will discuss two algorithms that are faster than the one proposed in Exercise 4.3.

4.1 The Algorithms of Euclid and Sugiyama

The *Euclidean algorithm* is a well-known algorithm that can be used to compute the *greatest common divisor* of two univariate polynomials. We assume that the reader is familiar with this algorithm. In order to fix notation, suppose we want to compute $\gcd(r_{-1}(Z), r_0(Z))$. Then the Euclidean algorithm proceeds as follows:

$$\begin{array}{rclcllcl}
r_{-1}(Z) & = & q_1(Z)r_0(Z) & + & r_1(Z), & \deg(r_1) & < & \deg(r_0) \\
r_0(Z) & = & q_2(Z)r_1(Z) & + & r_2(Z), & \deg(r_2) & < & \deg(r_1) \\
 & \vdots & & \vdots & & & \vdots & \\
r_{j-2}(Z) & = & q_j(Z)r_{j-1}(Z) & + & r_j(Z), & \deg(r_j) & < & \deg(r_{j-1}) \\
r_{j-1}(Z) & = & q_{j+1}(Z)r_j(Z). & & & & &
\end{array}$$

From this we can conclude that $\gcd(r_{-1}(Z), r_0(Z)) = r_j(Z)$. The key equation can be solved with the algorithm of *Sugiyama* in the following way.

Algorithm 4.6. Set

$$r_{-1}(Z) = Z^{\delta-1}, \quad r_0(Z) = S(Z), \quad U_{-1}(Z) = 0, \quad U_0(Z) = 1,$$

and proceed with the algorithm of Sugiyama until an $r_k(Z)$ is reached such that

$$\deg(r_{k-1}(Z)) \geq \frac{1}{2}(\delta - 1) \qquad \text{and} \qquad \deg(r_k(Z)) \leq \frac{1}{2}(\delta - 3),$$

also updating

$$U_i(Z) = q_i(Z)U_{i-1}(Z) + U_{i-2}(Z).$$

Then the error-locator and evaluator polynomial are

$$\begin{aligned} \sigma(Z) &= \epsilon U_k(Z), \\ \omega(Z) &= (-1)^k \epsilon r_k(Z), \end{aligned}$$

where ϵ is chosen such that $\sigma_0 = \sigma(0) = 1$.

Exercise 4.7. Show that the $\sigma(Z)$ and $\omega(Z)$ resulting from the algorithm satisfy

1. $\omega(Z) = \sigma(Z)S(Z) \bmod Z^{\delta-1}$,
2. $\deg(\sigma(Z)) \leq \frac{1}{2}(\delta - 1)$,
3. $\deg(\omega(Z)) \leq \frac{1}{2}(\delta - 3)$.

We will not prove the correctness of the algorithm. Sugiyama's algorithm is used for decoding in Project 7 on Golay codes.

4.2 The Algorithm of Berlekamp-Massey

The *Berlekamp-Massey algorithm* is an example of *dynamic programming* The algorithm is iterative, and in the j-th iteration the following problem is solved: find the pair $(\sigma_j(Z), \omega_j(Z))$ such that

1. $\sigma_j(0) = 1$,
2. $\sigma_j(Z)S(Z) = \omega(Z) \bmod Z^j$,
3. $d_j = \max\{\deg(\sigma_j), \deg(\omega_j) + 1\}$ is minimal.

It is rather technical to work out what has to be updated when proceeding to the next iteration. After the algorithm we will give a few remarks on the variables that are used.

Algorithm 4.8.

1. $j = 0$; $\quad \sigma_0 = -\omega'_0 = 1$; $\quad \sigma'_0 = \omega_0 = 0$; $\quad d_0 = 0$; $\quad \Delta = 1$.
2. $\boldsymbol{\Delta_j}$ **= coefficient of** $\boldsymbol{Z^j}$ **in** $\boldsymbol{\sigma_j(Z)S(Z) - \omega_j(Z)}$**.**

```
3. If $\Delta_j = 0$ then
   $d_{j+1} := d_j; \quad \sigma_{j+1} := \sigma_j; \quad \omega_{j+1} := \omega_j;$
   $\sigma'_{j+1} := Z\sigma'_j; \quad \omega'_{j+1} := Z\omega'_j$
4. If $\Delta_j \neq 0$ and $2d_j > j$ then
   $d_{j+1} := d_j; \quad \sigma_{j+1} := \sigma_j - \Delta_j\Delta^{-1}\sigma'_j; \quad \omega_{j+1} := \omega_j - \Delta_j\Delta^{-1}\omega'_j;$
   $\sigma'_{j+1} := Z\sigma'_j; \quad \omega'_{j+1} := Z\omega'_j$
5. If $\Delta_j \neq 0$ and $2d_j \leq j$ then
   $d_{j+1} := j + 1 - d_j; \quad \sigma_{j+1} := \sigma_j - \Delta_j\Delta^{-1}\sigma'_j; \quad \omega_{j+1} := \omega_j - \Delta_j\Delta^{-1}\omega'_j;$
   $\Delta := \Delta_j; \quad \sigma'_{j+1} := Z\sigma_j; \quad \omega'_{j+1} := Z\omega_j$
6. If $S_{j+1}$ is known then $j := j + 1$ and go to step 2; otherwise stop.
```

In the algorithm, the variables σ'_j and ω'_j are auxiliary. The Δ_j measures how far a solution to the j-th iteration is from being a solution to the $(j+1)$-th iteration. If $\Delta_j = 0$, the solution passes to the next iteration. If $\Delta_j \neq 0$, then the solution must be adjusted in such a way that the resulting $d_{j+1} = \max\{\deg(\sigma_{j+1}), \deg(\omega_{j+1}) + 1\}$ is minimal. In order to minimize this degree, the two cases 4 and 5 have to be distinguished.

Notice that in the algorithm of Sugiyama the degree of the polynomial decreases during the algorithm, whereas in the Berlekamp-Massey algorithm the degree of the polynomial increases. This is an advantage, since error-vectors of small weight are more likely to occur than those of high weight.

5. Gröbner Bases and Arbitrary Linear Codes

We will start by a general construction of a code, and later show that in fact this gives all linear codes.

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\} \subseteq \mathbb{F}_q^m$ be the set of zeros of a set of polynomials $\mathcal{G} = \{G_1, \ldots, G_u\}$ in $\mathbb{F}_q[X_1, X_2, \ldots, X_m]$. Let I be the ideal generated by $\mathcal{G}$. Define the ring R as

$$R = \mathbb{F}_q[X_1, \ldots, X_m]/I.$$

Let $F_1, F_2, \ldots, F_r$ be a basis of the $\mathbb{F}_q$-vector subspace L of R. Consider the evaluation map

$$ev_{\mathcal{P}} : L \longrightarrow \mathbb{F}_q^n.$$

The codes we consider here are

$$C = Im(ev_{\mathcal{P}})^{\perp}.$$

Thus $H = (F_i(P_j))$ is a parity check matrix of C. After introducing this algebraic setting, it is clear how Gröbner bases can be used for the decoding problem. Let d be the minimum distance of C. Suppose we receive a vector $\mathbf{y}$ and we want to decode t errors, with $t \leq \lfloor (d-1)/2 \rfloor$. Then, after computing the syndromes

$$s_i = \sum_{j=1}^{n} y_j F_i(P_j),$$

we can form the following system of equations $\mathcal{S}(\mathbf{s}, v)$:

$$\begin{cases} \sum_{j=1}^{v} Y_j F_i(X_{1j}, \dots, X_{mj}) & = & s_i & \text{for } i = 1, \dots, r \\ G_i(X_{1j}, \dots, X_{mj}) & = & 0 & \text{for } j = 1, \dots, v \text{ and } i = 1, \dots, u \\ Y_j^q & = & Y_j & \text{for } j = 1, \dots, t, \end{cases}$$

with variables $X_{1j}, \dots, X_{mj}$ for the coordinates of a copy of $\mathbb{F}_q^m$ for all $j = 1, \dots, v$, and the variables $Y_1, \dots, Y_v$ for the error values in $\mathbb{F}_q$. As in the case of cyclic codes, we see that if $(\mathbf{x}_1, \dots, \mathbf{x}_v, \mathbf{y}_1, \dots, \mathbf{y}_v)$, with $\mathbf{x}_j = (x_{1j}, \dots, x_{mj})$, is a solution to $\mathcal{S}(\mathbf{s}, v)$, then so is

$$(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(v)}, \mathbf{y}_{\pi(1)}, \dots, \mathbf{y}_{\pi(v)}),$$

for any permutation π of $\{1, \dots, v\}$. Hence a Gröbner basis $\mathcal{G}$ for the ideal $\mathcal{S}(\mathbf{s}, t)$ with respect to the lexicographic order with

$$Y_m > \cdots > Y_1 > X_{mv} > \cdots > X_{1v} > \cdots > X_{m1} > \cdots > X_{11}$$

will have elements that are polynomials in $X_{m1}, \dots, X_{11}$ only. These elements generate the ideal $\mathcal{S}(\mathbf{s}, v) \cap \mathbb{F}_q[X_{11}, \dots, X_{m1}]$. This intersection has no solution when $v < t$. If $v = t$, then the intersection is the *error-locator ideal*, that means that it has the set of error positions as zeroset in $\mathbb{F}_q^m$. The error values can be found as before for cyclic codes with Proposition 2.1.

Example 5.1. Let C be an $[n, k, d]$ linear code with $r \times n$ parity check matrix H, where $r = n - k$. Consider the n columns of H as points $P_1, \dots, P_n \in \mathbb{F}_q^r$ and set $\mathcal{P} = \{P_1, \dots, P_n\}$. Then $\mathcal{P}$ is finite, so it is an algebraic set:

$$\mathcal{P} = \mathcal{Z}_{\mathbb{F}_q}(I), \quad I = \{G \in \mathbb{F}_q[X_1, \dots, X_r] \mid G(P_1) = \cdots = G(P_n) = 0\}.$$

If we take as an r-dimensional vector space L the coordinate functions

$$L = \langle X_1, \dots, X_r \rangle,$$

then it is clear that $C = Im(ev_{\mathcal{P}})^{\perp}$.

Exercise 5.2. Describe the Hamming code by the above method. What is the vanishing ideal in $\mathbb{F}_2[X_1, X_2, X_3]$ if one applies the above procedure to the Hamming code?

Although in principle every linear code could be described and decoded in this way, the large number of variables will make it very impractical. The following exercise relaxes the number of variables a bit.

Exercise 5.3. Let C be a q-ary $[n, k, d]$ code. Let $r = n - k$. Let $H = (h_{ij})$ be a parity check matrix of C. Let m be a positive integer such that $q^m \geq n$. Show that there exist n distinct points $P_1, \dots, P_n$ in $\mathbb{F}_q^m$ and polynomials $F_1, \dots, F_r$ in $\mathbb{F}_q[X_1, \dots, X_m]$ such that $F_i(P_j) = h_{ij}$.

Example 5.4. Let C be a cyclic code with defining set J. Instead of treating this as an arbitrary linear code as in the previous example, it is better to use the structure of the parity check matrix, as follows. Take $\mathcal{P} = \{1, \alpha, \ldots, \alpha^{n-1}\} \subseteq \mathbb{F}_{q^e}$, the set of n-th roots of unity. Hence

$$I = (X^n - 1)\mathbb{F}_{q^e}[X].$$

If we take for L the vector space

$$L = \langle X^j \mid j \in J \rangle$$

over $\mathbb{F}_{q^e}$, it is clear that C is a code as described above, and that the system $\mathcal{S}(\mathbf{s}, t)$ we have to solve, equals the one we already met in Section 3.

One-step decoding is done in the same way as for cyclic codes by treating the s_j as variables and the corresponding Theorem 3.17 holds.

The same methods applies for getting the minimal weight codewords of a linear code.

Notes

That the general decoding problem is hard can be made precise in terms of complexity theory. See [3, 5].

Formulas for the probability of a decoding error or failure for several decoders and the relation with the weight enumerator is given in [6, 24]. Some history of the origins of decoding algorithms can be found in [2].

The original idea of one-step decoding is from [9, 10] and [30]. See also [38].

The method to decode cyclic codes up to half the actual minimum distance using Gröbner bases is from [11, 12, 13]. The extension to arbitrary linear codes is from [17, 18]. Theorem 3.17 is from [17, 18, 25]. The remark in Exercise 3.11 is from [25]. In this paper the work of [15] is used to transform a Gröbner basis of a zero-dimensional ideal with respect to one reduction order into a Gröbner basis with respect to another one. The decoding is considerably faster by this method as is seen in the Project on the Golay code. Decoding constacyclic codes in Lee metric by the use of Gröbner bases is explained in [28].

A more efficient way to decode cyclic codes is by solving the key equation [1, 4, 20, 27, 31, 37]. The formula for the error values is from [19].

The material of Section 4 is from [6, 7, 26, 32]. This formulation of the Berlekamp-Massey algorithm is from [14].

For Reed-Solomon codes a hybrid of the algorithm of Berlekamp-Massey and Gröbner bases techniques is given in [39, 40, 41] to get all closest codewords of a received word.

Decoding arbitrary linear codes with Gröbner bases is from [17, 18]. This method can also be applied to get all minimal weight codewords as explained for cyclic codes in Chapter 10.

There are many papers on decoding algebraic geometry codes and we refer to the literature [8, 16, 21, 22, 23, 29].

The Berlekamp-Massey algorithm is generalized to polynomials in several variables by [34, 35, 36]. This theory has very much to do with the theory of Gröbner bases, but it solves another problem than Buchberger's algorithm. The algorithm is implemented in the decoding of algebraic geometry codes. See the literature cited above and [33]. The name *footprint* for the Δ-set is from [8].

References

1. S. Arimoto (1961): *Encoding and decoding of p-ary group codes and the correction system*, (in Japanese) Inform. Processing in Japan **2**, 320–325.
2. A. Barg (1993): *At the dawn of the theory of codes*, Math. Intelligencer **15**, 20–27.
3. A. Barg: *Complexity issues in coding theory*, to appear in Handbook of Coding Theory, (V.S. Pless, W.C. Huffman and R.A. Brualdi eds.), Elsevier.
4. E.R. Berlekamp (1984): *Algebraic Coding Theory*, Aegon Park Press, Laguna Hills CA.
5. E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg (1978): *On the inherent intractibility of certain coding problems*, IEEE Trans. Inform. Theory **24**, 384–386.
6. R.E. Blahut (1983): *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading.
7. R.E. Blahut (1985): *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, Reading.
8. R.E. Blahut, *Introduction to Algebraic Coding*, book in prepartation.
9. A. Brinton Cooper III (1990): *Direct solution of BCH decoding equations*, Communication, Control and Signal Processing, Elsevier Sc. Publ., 281–286.
10. A. Brinton Cooper III (1991): *Finding BCH error locator polynomials in one step*, Electronic Letters **27**, 2090–2091.
11. X. Chen, I.S. Reed, T. Helleseth, and T.K. Truong (1994): *Algebraic decoding of cyclic codes: a polynomial point of view*, Contemporary Math. **168**, 15–22.
12. X. Chen, I.S. Reed, T. Helleseth, and T.K. Truong (1994): *Use of Gröbner bases to decode binary cyclic codes up to the true minimum distance*, IEEE Trans. Inform. Theory **40**, 1654–1661.
13. X. Chen, I.S. Reed, T. Helleseth, and T.K. Truong (1994): *General principles for the algebraic decoding of cyclic codes*, IEEE Trans. Inform. Theory **40**, 1661–1663.
14. J.L. Dornstetter (1987): *On the equivalence of Berlekamp's and Euclid's algorithm*, IEEE Trans. Inform. Theory **33**, 428–431.
15. J.C. Faugère, P. Gianni, D. Lazard, and T. Mora (1993): *Efficient computation of zero-dimensional Gröbner bases by a change of ordering*, Journ. Symb. Comp. **16**, 329–344.
16. G.-L. Feng and T.R.N. Rao (1993): *Decoding of algebraic geometric codes up to the designed minimum distance*, IEEE Trans. Inform. Theory **39**, 37–45.
17. J. Fitzgerald (1996): *Applications of Gröbner bases to linear codes*, Ph.D. Thesis, Louisiana State Univ.
18. J. Fitzgerald and R.F. Lax (1998): *Decoding affine variety codes using Gröbner bases*, Designs, Codes and Cryptography **13**, 147–158.
19. G.D. Forney Jr. (1965): *On decoding BCH codes*, IEEE Trans. Inform. Theory **11**, 549–557.
20. D.C. Gorenstein and N. Zierler (1961): *A class of error-correcting codes in p^m symbols*, Journ. SIAM **9**, 207–214.
21. T. Høholdt, J.H. van Lint, and R. Pellikaan, *Algebraic geometry codes*, to appear in Handbook of Coding Theory, (V.S. Pless, W.C. Huffman and R.A. Brualdi eds.), Elsevier.
22. T. Høholdt and R. Pellikaan (1995): *On decoding algebraic-geometric codes*, IEEE Trans. Inform. Theory **41**, 1589–1614.

23. J. Justesen, K. J. Larsen, H. Elbrønd Jensen, A. Havemose, and T. Høholdt (1989): *Construction and decoding of a class of algebraic geometric codes*, IEEE Trans. Inform. Theory **35**, 811–821.
24. T. Kløve and V. I. Korzhik (1995): *Error Detecting Codes*, Kluwer Acad. Publ., Dordrecht.
25. P. Loustaunau and E. V. York (1997): *On the decoding of cyclic codes using Gröbner bases*, AAECC **8**, 469–483.
26. F. J. MacWilliams and N. J. A. Sloane (1977): *The Theory of Error-Correcting Codes*, North-Holland Math. Library **16**, North-Holland, Amsterdam.
27. J. L. Massey (1969): *Shift-register synthesis and BCH decoding*, IEEE Trans. Inform. Theory **15**, 122–127.
28. J. Maucher and R. Kötter (1996): *Decoding constacyclic codes in Lee- and Mannheim metric by the use of Gröbner bases*, preprint.
29. R. Pellikaan (1993): *On the efficient decoding of algebraic-geometric codes*, pp. 231–253 in Proceedings of Eurocode 92, CISM Courses and Lectures **339**, Springer-Verlag, Wien New York.
30. W. T. Penzhorn (1993): *On the fast decoding of binary BCH codes*, pp. 103 in Proc. IEEE Int. Symp. Inform. Theory, San Antonio.
31. W. W. Peterson (1960): *Encoding and error-correction procedures for the Bose-Chauduri codes*, IRE Trans. Inform. Theory **6**, 459–470.
32. W. W. Peterson and E. J. Weldon (1977): *Error-Correcting Codes*, MIT Press, Cambridge.
33. K. Saints and C. Heegard (1995): *Algebraic-geometric codes and multidimensional cyclic codes: A unified theory and algorithms for decoding using Gröbner bases*, IEEE Trans. Inform. Theory **41**, 1733–1751.
34. S. Sakata (1981): *On determining the independent point set for doubly periodic arrays and encoding two-dimensional cyclic codes and their duals*, IEEE Trans. Inform. Theory **27**, 556–565.
35. S. Sakata (1988): *Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array*, Journal of Symbolic Computation 5, 321–337.
36. S. Sakata (1990): *Extension of the Berlekamp-Massey algorithm to N dimensions*, Information and Computation **84**, 207–239.
37. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa (1975): *A method for solving the key equation for decoding Goppa codes*, Information and Control **27**, 87–99, 1975.
38. H.-J. Weber (1994): *Algebraische Algorithmen zur Dekodierung zyklischer Codes*, Master's Thesis, Univ. Dortmund.
39. D-J. Xin (1993): *New approach to decoding Reed-Solomon codes based on generalized rational interpolation*, pp. 219–223 in Proc. Sixth Swedish-Russian International Workshop Inform. Trans.
40. D-J. Xin (1994): *Homogeneous interpolation problem and key equation for decoding Reed-Solomon codes*, Science in China (Series A) **37**, No. 11.
41. D-J. Xin (1995): *Extension of the Welch-Berlekamp theorem and universal strategy of decoding algorithm beyond BCH bound*, Science in China (Series A) **38**, No. 11.

Project 1. Automatic Geometry Theorem Proving

Tomas Recio, Hans Sterk, and M. Pilar Vélez

1. Introduction

The aim of this project is to illustrate how the framework of polynomial rings and computational methods designed for them can be of help in proving (plane) geometry theorems. The idea is not original and there are already, even for the beginner, excellent references concerning this topic. In coherence with the 'tapas' style of this book, we recall a few, tasty ones: for instance, the recent book by the founder of the modern approach to automatic geometry theorem proving, Wu Wen Tsun [5]; the textbook [2], which integrates one section on this material in a commutative algebra/algebraic geometry course, and the book by Chou [1], including an impressive collection of computed examples.

The primary motivation for this contribution has been the preparation of undergraduate classroom material for computer-aided commutative algebra courses that have been offered, since the middle eighties, at the University of Cantabria and, more recently, at the University Complutense of Madrid. Thus, the following pages should be regarded as an elaborated version of teaching notes; a preliminary version of the notes was used for a Galois/Eidma course at the Eindhoven University of Technology. The rationale of our didactical approach is that algebraic geometry examples improve students' understanding of commutative algebra concepts and conversely.

Automatic geometry theorem proving provides an interesting framework to accomplish this, since an elementary geometry problem has to be modeled into a commutative algebra statement, which will be, in turn, regarded as a property of algebraic varieties. In this way students develop the computational skills in commutative algebra to decide on the status (true or false) of elementary geometry statements. The didactical relevance is that, in the context of elementary geometry theorems, the students' 'a priori' intuition is confronted with the actual behaviour of mathematical objects; this confrontation seems the key to significant learning.

As a consequence of the didactical origin of the chapter, our classroom presentation of the topic turned out to converge towards the style of the book [2], several years before its publication. This coincidence reflects the obvious fact that Gröbner bases are very likely to be introduced in most computationally oriented commutative algebra courses and it is also due to a common exploitation of Kapur's [3] formulation. We thank the authors of [2] for sending us an earlier draft of their manuscript. Some results below are similar to theirs, but we take full responsibility for many deviations and interpretations. Besides, we have enlarged their presentation to include an introduction to automatic *discovery* of theorems. In other words, we proclaim automatization not only for proving a result, but even for inventing results!

Given the complexity of current algorithms for ideal manipulation via Gröbner bases and the usually limited computing resources available in undergraduate mathematics laboratories, it is not straightforward to identify a collection of examples that can be successfully manipulated with scientific freeware, such as CoCoA[1], running on small machines. We hope this chapter also shows how some interesting instances of automatic geometry theorem proving are tractable with Gröbner bases, despite the common belief that they require the more standard approach via characteristic sets.

2. Approaches to Automatic Geometry Theorem Proving

Although there are several possible approaches to automatic geometry theorem proving, the main steps are always similar:

1) *Algebraic formulation*: the translation of a geometry statement into algebraic equations.
2) *Proof:* the use of some decision procedure, in the model we are working with, to determine the validity of the theorem.
3) *Searching conditions:* the search for extra conditions if the theorem, as it was formulated originally, is false.

This project is organized around these items; it is a tour along classical results from geometry, with an illustration of the peculiarities that may arise. In the examples we will use the computer algebra package CoCoA, but the computations can also be done with various other systems.

3. Algebraic Geometry Formulation

Let K be a field of characteristic 0, for instance the field of rational numbers $\mathbb{Q}$, and let L be an algebraically closed field containing K, for instance the field of complex numbers $\mathbb{C}$. We will restrict our attention to plane geometry theorems which can be phrased in terms of polynomial equalities over K.

For the rest of this paper, variety means K-variety, open set means K-open, etc.

[1] **CoCoA is scientific software, produced and freely distributed by Robbiano-Niesi-Capani, Universitá di Genoa, cocoa@dima.unige.it**

Start by choosing an appropriate coordinate system. Variables $\mathbf{x} = (x_1, \dots, x_d)$, used to describe coordinates of points or geometric magnitudes (distance, radius, etc.) that can be chosen arbitrarily, are called *independent variables*[2]; variables $\mathsf{y} = (y_1, \dots, y_r)$, used to describe points that satisfy certain equations in the independent ones because of the construction procedure, are called *dependent variables.* In this manner, various geometric statements such as incidence, parallelism, perpendicularity, distance, etc., can be turned into polynomial equations in the variables $(\mathbf{x}, \mathsf{y})$ with coefficients in K.

Example 3.1. $\mathsf{ab} \perp \mathsf{cd}$ translates into

$$(b_1 - a_1)(d_1 - c_1) + (b_2 - a_2)(d_2 - c_2) = 0,$$

where $\mathsf{a} = (a_1, a_2)$, $\mathsf{b} = (b_1, b_2)$, etc.

The midpoint of ab is described by the two equations

$$2u_1 = a_1 + b_1 \text{ and } 2u_2 = a_2 + b_2.$$

Here, u_1, u_2 are dependent variables.

Remark 3.2. In our translations we agree not to take advantage of special features of a particular construction. For instance, we translate parallelism of ab and cd into $(a_1 - b_1)(c_2 - d_2) - (a_2 - b_2)(c_1 - d_1) = 0$. If a happens to be $(a_1, 0)$ and b happens to be $(b_1, 0)$, then we specialize this expression to $(a_1 - b_1)(c_2 - d_2) = 0$ instead of using the particular form $c_2 - d_2 = 0$.

Exercise 3.3. Express the following conditions as polynomial equations.

1. The point a lies on a circle with center m and radius r.
2. The point a lies on the line bc through points b and c.
3. Points a, b, and c are collinear, i.e., on one line.

After adopting a coordinate system, the *hypotheses* of a theorem can, by assumption, be expressed as a set of polynomial equations, $h_1(\mathbf{x}, \mathsf{y}) = 0, \dots, h_p(\mathbf{x}, \mathsf{y}) = 0$, and the *thesis* can be expressed as a polynomial equation, $t(\mathbf{x}, \mathsf{y}) = 0$, where $h_1, \dots, h_p, t \in K[\mathbf{x}, \mathsf{y}]$. A geometry theorem $\mathcal{T}$ is translated into

$$\forall (\mathbf{x}, \mathsf{y}) \in L^n \quad h_1(\mathbf{x}, \mathsf{y}) = 0, \dots, h_p(\mathbf{x}, \mathsf{y}) = 0 \Rightarrow t(\mathbf{x}, \mathsf{y}) = 0, \tag{3.1}$$

where $n = d + r$. In terms of algebraic geometry, this is phrased as: the algebraic variety defined by $\{t = 0\}$ contains $\{h_1 = 0, \dots, h_p = 0\} \subset L^n$.

At this point we need to introduce some notation from algebraic geometry: given $f_1, \dots, f_q \in K[\mathbf{x}, \mathsf{y}]$ we denote by $\mathcal{Z}(f_1, \dots, f_q) \subset L^n$ the algebraic variety (or set) defined by $f_1, \dots, f_q$ in L^n; given an algebraic variety $Z \subset L^n$ we denote by $I(Z)$ the ideal defined by Z in $K[\mathbf{x}, \mathsf{y}]$ (cf. Chapter 1).

[2] **This is a subtle point to which we will come back in Section 4.**

Definition 3.4. Given a geometry theorem $\mathcal{T}$, we define the *hypotheses variety* H as the algebraic set $\mathcal{Z}(h_1, \ldots, h_p)$ and the *thesis variety* T as the algebraic set $\mathcal{Z}(t)$. The ideal $\sqrt{(h_1, \ldots, h_p)}$ is called the *hypotheses ideal.*

Definition 3.5. A theorem $\mathcal{T}$ is *geometrically true* if the hypotheses variety H is contained in the thesis variety T.

The notion of being geometrically true is related to the ideal membership problem in the following way.

Theorem 3.6. *The following statements are equivalent:*

(a) Theorem $\mathcal{T}$ is geometrically true.
(b) $t \in \sqrt{(h_1, \ldots, h_p)}$.
(c) $1 \in (h_1, \ldots, h_p, tz - 1)K[\mathbf{x}, \mathbf{y}, z]$.

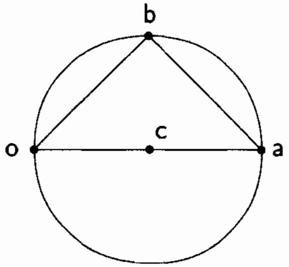
Exercise 3.7. Show that *(b)* and *(c)* are always equivalent, i.e., do not need the assumption that L be algebraically closed. Indicate where you use Hilbert's Nullstellensatz in the proof of the above theorem.

Item *(c)* of the theorem is suitable for the use of a computer algebra system, such as CoCoA. In CoCoA, $\mathsf{NormalForm}(f, (f_1, \ldots, f_q))$ computes the normal form of the polynomial f with respect to a Gröbner basis of the ideal generated by $\{f_1, \ldots, f_q\}$. Of course, we have to select an ordering of the variables, but since we are only interested in deciding if the normal form is or is not 0 – and this is independent of the ordering – it makes sense to choose an ordering such as DegRevLex, which has the reputation of allowing faster computations. In conclusion, we have

$$\mathsf{NormalForm}(1, (h_1, \ldots, h_p, tz - 1)) \begin{cases} = 0 & \mathcal{T} \text{ is geometrically true} \\ \neq 0 & \mathcal{T} \text{ is not geometrically true} \end{cases}$$

It is important to remark that there is no unique algebraic formulation for a given geometric statement. When we talk about proving a theorem $\mathcal{T}$, we implicitly refer to the selected algebraic translation. In particular, it is often useful to choose formulations that reduce the number of variables appearing in the statement. For example, since most geometric properties are invariant under similarities, one can often translate a given theorem into an equivalent statement in which one or several points have been assigned numerical coordinates. Here is a simple but illustrative example.

Example 3.8. The angle subtended by a diameter of a circle from any point on the circumference is a right angle.



This statement concerns any circle and any point on it. But it is obvious that the theorem is true in general if and only if it is true for one concrete circle (since any two circles are similar and similarities preserve right angles). Thus we can fix (totally or partially) the given circle. Let us fix the center but not the radius. Take points o $= (0,0)$, a $= (2l, 0)$ and b $= (u, v)$ such that the segment between o and a is a diameter of a circle and b belongs to this circle. Observe that l, u, v are the variables, that l, u can be considered as independent and that v can be considered as dependent on l, u since it satisfies the equation of the circle.

Hypothesis: the fact that b is on the circle centered at $(l, 0)$ with radius l translates into

$$h = h(l,u,v) = (u-l)^2 + v^2 - l^2 = 0.$$

Thesis: the angle $\widehat{\mathsf{oba}}$ is a right angle, i.e., ba $\perp$ bo,

$$t = t(l,u,v) = u(u-2l) + v^2.$$

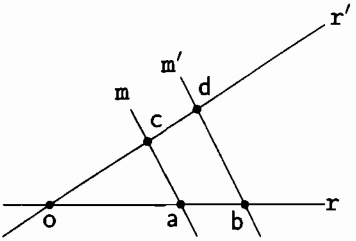
Thus, we must check whether NormalForm$(1, (h, tz-1)) = 0$, which is easily verified in CoCoa. Therefore, the theorem is geometrically true. (Of course, the computation in this example is trivial, even by hand.)

Exercise 3.9. Describe hypotheses and theses in the following cases and show that the two statements are geometrically true.

1. In a right triangle oba with right angle at b, let p be the projection of b on oa. Then
$$\frac{|\mathsf{oa}|}{|\mathsf{ob}|} = \frac{|\mathsf{ob}|}{|\mathsf{op}|}.$$
2. Same situation as before. Then
$$\frac{|\mathsf{op}|}{|\mathsf{bp}|} = \frac{|\mathsf{bp}|}{|\mathsf{pa}|}.$$

It seems that we have found a nice way to prove geometry theorems. Unfortunately, there are well-known theorems that seem 'false' using this method. For instance, according to this procedure, Thales' Theorem turns out to be not geometrically true as the following example shows.

Example 3.10. (Thales' Theorem) Given two secant lines r and r′, the triangles obtained by intersecting any two parallel lines m and m′ with the two secants are similar.



Consider the x-axis as one of the secant lines and the line joining points $\mathsf{o} = (0, 0)$ and $\mathsf{c} = (p, q)$ as the other one. Take points $\mathsf{a} = (l, 0)$ and $\mathsf{b} = (s, 0)$ on the x-axis and draw the line ac. Let $\mathsf{d} = (u, v)$ be the intersection of oc and the line parallel to ac passing through b.

Hypotheses: $\mathsf{d} \in \mathsf{oc}$: $h_1(l, s, p, q, u, v) = qu - pv = 0$

$\mathsf{ac} \parallel \mathsf{bd}$: $h_2(l, s, p, q, u, v) = q(u - s) - v(p - l) = 0$

Thesis: the ratios of the lengths of the corresponding sides of the two triangles oac and obd are equal, i.e.,

$$\frac{|\mathsf{oa}|}{|\mathsf{ob}|} = \frac{|\mathsf{oc}|}{|\mathsf{od}|} = \frac{|\mathsf{ac}|}{|\mathsf{bd}|}.$$

This is expressed by the following equations: $t_1 := (u^2+v^2)l^2 - s^2(p^2+q^2) = 0$, $t_2 := ((s - u)^2 + v^2)l^2 - s^2((p - l)^2 + q^2) = 0$.

We must check that the hypotheses variety $\{h_1 = 0, h_2 = 0\}$ is contained in the zeroset of t_1 (respectively, t_2). CoCoA's answer for thesis t_1 is negative, so this thesis is not geometrically true:

```
Ring ( "ring name:" R ; "characteristic:" 0 ;
     "variables:" zuvpqsl ; "weights:" 1 , 1 , 1 , 1 , 1 , 1 , 1
     "ordering:" DEGREVLEX );
```

$$\mathsf{NormalForm}(1, \mathsf{Ideal}(-vp + uq, -vp + uq - qs + vl, \\ - zp^2s^2 - zq^2s^2 + zu^2l^2 + zv^2l^2 - 1));$$

```
 1
```

A similar computation shows that the second thesis is not geometrically true.

This last example makes clear that our procedure to prove geometry theorems is not complete: if the answer is YES we can guarantee the statement's validity, but if the answer is NO the theorem can still be 'true'. This can happen if our algebraic formulation does not correctly represent the geometric construction we have in mind. For example, in proving Thales' theorem (Example 3.10) by hand, it is necessary at some point to avoid degenerate cases (e.g., the case where c is on the x-axis). In algebraic terms this means that certain expressions should not assume the value 0. Nevertheless, these degenerate cases satisfy the algebraic hypotheses, but the theorem may not hold for all these cases. Let us deal with this problem.

Let $\mathcal{T}$ be a geometry theorem and suppose that it is not geometrically true, i.e., $H \not\subset T$. The validity of the theorem, however, can be thought of as a *generic* matter in the following sense: it can happen that for some polynomial $g \in K[\mathbf{x}, \mathbf{y}]$, the smaller set $H \setminus \mathcal{Z}(g)$ is contained in T, i.e., upon removing some degenerate cases from the hypotheses variety, the thesis holds over the remaining configurations. Therefore, we propose a change in the formulation of (3.1), which allows for imposing a condition:

$$\forall(\mathbf{x}, \mathbf{y}) \in L^n \quad h_1(\mathbf{x}, \mathbf{y}) = 0, \ldots, h_p(\mathbf{x}, \mathbf{y}) = 0, g(\mathbf{x}, \mathbf{y}) \neq 0 \Rightarrow t(\mathbf{x}, \mathbf{y}) = 0. \quad (3.2)$$

Definition 3.11. Let $h_1, \ldots, h_p, g, t \in K[\mathbf{x}, \mathbf{y}]$ as above. We define the *hypotheses+condition variety* H_g as the algebraic set $\mathcal{Z}(h_1, \ldots, h_p, gk-1)$ in L^{n+1}, where k is a new indeterminate.

Definition 3.12. Let $h_1, \ldots, h_p, g, t \in K[\mathbf{x}, \mathbf{y}]$. A theorem of the form (3.2) is *geometrically true under the condition* $g \neq 0$ if the hypotheses+condition variety H_g is contained in the thesis variety $T = \mathcal{Z}(t) \subset L^{n+1}$.

Exercise 3.13. Show that the validity of a theorem under the condition $g \neq 0$ is equivalent to $t \in \sqrt{(h_1, \ldots, h_p, gk - 1)}$. Prove that this last condition holds if and only if

$$1 \in (h_1, \ldots, h_p, gk - 1, tz - 1)K[\mathbf{x}, \mathbf{y}, k, z],$$

where z is a new indeterminate.

Also show that, under the projection $L^{n+1} \to L^n$ on the first n coordinates, H_g is identified with the set $H \cap \{g \neq 0\}$.

Now let us go back to Example 3.10.

Exercise 3.14. (Thales' Theorem revisited) We proved above that, without any extra condition, Thales' Theorem is not geometrically true. Now impose the first nondegeneracy condition that arises, namely that the line oc be different from the x-axis (i.e., $q \neq 0$). Check that the theorem is geometrically true under this condition.

The next section shows how to look for such nondegeneracy conditions.

4. Searching for Conditions

Notation remains as in the previous section, i.e., $h_1, \ldots, h_p$ describe the hypotheses and t the thesis for a geometry theorem. Our first goal is to investigate single conditions under which a geometry theorem becomes true.

Definition 4.1. A *nondegeneracy condition* for a geometry theorem is a polynomial $g \in K[\mathrm{x}, \mathrm{y}]$ such that the theorem is geometrically true under the condition $g \neq 0$.

Exercise 4.2. Prove that a polynomial $g \in K[\mathrm{x}, \mathrm{y}]$ is a nondegeneracy condition for a geometry theorem if and only if

$$g^l \in (h_1, \dots, h_p, tz - 1) \cap K[\mathrm{x}, \mathrm{y}]$$

for some $l \geq 0$.

Remark 4.3. From the computational point of view, it is easier to search for conditions among the elements of the ideal $(h_1, \dots, h_p, tz - 1) \cap K[\mathrm{x}, \mathrm{y}]$ than among the elements of its radical. Radical computation is more difficult and less often implemented in computer algebra packages. However, for our geometric purposes it makes no difference because $g(p) \neq 0$ if and only if $g^l(p) \neq 0$.

The last remark motivates the following

Definition 4.4. The ideal

$$(h_1, \dots, h_p, tz - 1) \cap K[\mathrm{x}, \mathrm{y}]$$

will be called the *ideal of nondegeneracy conditions* for the given theorem $\mathcal{T}$.

This definition is too coarse in the sense that there exist conditions that make no sense for our purposes. For example, if the set $\mathcal{Z}(g)$ contains the hypotheses variety, then $H_g = \emptyset$ and, logically, any thesis follows from H_g; this occurs if $g \in \sqrt{(h_1, \dots, h_p)}$. If, at the other extreme, the set $\{g \neq 0\}$ contains H, then g does not really impose a condition: $H_g = H$.

In order to avoid such situations, we classify conditions as follows.

Definition 4.5. Let $g \in k[\mathrm{x}, \mathrm{y}]$ be a condition for a geometry theorem.

(i) g is a *trivial condition* if $g \in \sqrt{(h_1, \dots, h_p)}$.
(ii) Otherwise, g is a *nontrivial condition* and we distinguish two cases:
  a) g is a *relevant condition* if $1 \notin (h_1, \dots, h_p, g)$.
  b) g is an *irrelevant condition* if $1 \in (h_1, \dots, h_p, g)$; in this case $H_g = H$.

Remark 4.6. Trivial or irrelevant conditions, though unimportant by themselves, do play a role because it can happen that relevant conditions arise as combinations of other conditions, including trivial and irrelevant ones.

Exercise 4.7. Let $\mathcal{T}$ be a geometry theorem which is not geometrically true. Prove the following statements:

1. **If the hypotheses ideal $\sqrt{(h_1, \dots, h_p)}$ is prime, all conditions are trivial.**
2. **If nontrivial conditions for $\mathcal{T}$ exist, they are all relevant.**

3. There are relevant conditions for $\mathcal{T}$ if and only if there are relevant conditions in any basis of the ideal of nondegeneracy conditions of $\mathcal{T}$.

Remark 4.8. The computation of the ideal of nondegeneracy conditions (see Definition 4.4) with CoCoA is done using the command

$$\mathsf{Elim}(z, \mathsf{Ideal}(h_1, \ldots, h_p, tz - 1)),$$

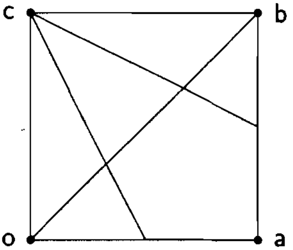
which yields a Gröbner basis of $(h_1, \ldots, h_p, tz - 1) \cap K[\mathrm{x}, \mathrm{y}]$.

We discard every element of this basis that represents a trivial condition, i.e., is contained in the hypotheses ideal $\sqrt{(h_1, \ldots, h_p)}$, in the following way: compute

$$\mathsf{NormalForm}(1, \mathsf{Ideal}(h_1, \ldots, h_p, gk - 1))$$

in $K[\mathrm{x}, \mathrm{y}, k]$; if it is 0 the condition is trivial. For nontrivial conditions we detect relevant and irrelevant ones by using NormalForm again.

Exercise 4.9. [3] Let oabc be a square. Then the two lines connecting c with the midpoints of oa and ab, respectively, divide the diagonal ob into three segments of equal length.



1. Use Exercise 3.3 to give a translation into a system of polynomial equations. Take $\mathsf{o} = (0,0), \mathsf{a} = (l,0), \mathsf{b} = (l,l), \mathsf{c} = (0,l)$.
2. Show that the theorem is not geometrically true.
3. Is the hypotheses ideal prime? If not, can you find a decomposition as intersection of prime ideals?
4. Analyze trivial and nontrivial conditions in the basis of the ideal of conditions.
5. Find a nondegeneracy condition so that the theorem holds under this extra condition.

Sets of the kind $H \setminus \mathcal{Z}(g)$ are Zariski open in the hypotheses variety H; as is well known, such sets form a basis for the topology on H: every open subset of H is a union of such special open sets.

[3] This theorem appears in the proposal 'School Mathematics in the 1990s' (ed. Geoffrey Howson and Bryan Wilson, Cambridge University Press, Cambridge, 1986) of the International Commission on Mathematical Instruction, where the didactical impact of automatic theorem proving in elementary geometry is already mentioned.

Exercise 4.10. Show that there is a nonempty Zariski open set in H where the thesis $t = 0$ holds if and only if there exists a nontrivial condition g such that: $h_1 = 0, \ldots, h_p = 0, g \neq 0 \Rightarrow t = 0$.

In searching for conditions, the set of 'failures' $\{t \neq 0\} \cap H$ plays a central role as the following exercise explains.

Exercise 4.11. Let $\{g_1, \ldots, g_s\}$ be a basis of the ideal of nondegeneracy conditions

$$(h_1, \ldots, h_p, tz - 1) \cap K[\mathrm{x}, \mathrm{y}].$$

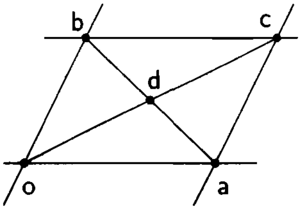
1. Prove that the algebraic set $\mathcal{Z}(g_1, \ldots, g_s)$ is the Zariski closure of $\{t \neq 0\}$ in H. Conclude that $\mathcal{Z}(g_1, \ldots, g_s)$ is the union of the irreducible components of H that meet $\{t \neq 0\}$.
2. Prove that, therefore, $\mathcal{Z}(g_1, \ldots, g_s)$ does not contain a nonempty Zariski open subset of H contained in $\mathcal{Z}(t)$.
3. Show that there is a proper algebraic set (possibly empty) W of H such that $\mathcal{Z}(g_1, \ldots, g_s) \setminus W \subset \{t \neq 0\} \cap H$.

For instance, in Exercise 4.9 the Zariski closure of $\{t \neq 0\} \cap H$ is equal to $\mathcal{Z}(l) \cap H$, the set of degenerate squares. The meaning of the first two items of Exercise 4.11 is that $l = 0$ is a necessary condition for the thesis to fail over some point of H. On the other hand, the third item shows that it may not be a sufficient condition: there could be some values of $\mathcal{Z}(l) \cap H$ where the thesis holds, but such values are contained in a proper Zariski-closed set of $H \cap \mathcal{Z}(l)$. Intuitively speaking, we could think of the set $\mathcal{Z}(g_1, \ldots, g_s)$ as the collection of truly degenerate cases where, perhaps, a few of these cases still satisfy the theorem.

Exercise 4.12. Find, in Exercise 4.9, the set of points in $\mathcal{Z}(l) \cap H$ that satisfy the thesis.

But, as you can see in the next example, sometimes $\mathcal{Z}(g_1, \ldots, g_s)$ contains all the 'usual' cases.

Example 4.13. Suppose we want to prove the following statement: The center of a parallelogram is on one of its edges.



Consider the parallelogram with vertices $\mathsf{o} = (0,0)$, $\mathsf{a} = (l,0)$, $\mathsf{b} = (r,s)$ and $\mathsf{c} = (p,q)$. Let $\mathsf{d} = (u,v)$ be the center of this parallelogram, i.e., the intersection of the diagonals. Here l, r, s are the independent variables.

Hypotheses: $\mathsf{oa} \parallel \mathsf{bc}$: $h_1 := l(s-q) = 0$
$\mathsf{ob} \parallel \mathsf{ac}$: $h_2 := qr - s(p-l) = 0$
$\mathsf{d} \in \mathsf{oc}$: $h_3 := uq - vp = 0$
$\mathsf{d} \in \mathsf{ab}$: $h_4 := s(u-l) - v(r-l) = 0$

Thesis: $\mathsf{d} \in \mathsf{oa}$: $t := lv = 0$

```
Ring ( "ring name:" R ; "characteristic:" 0 ;
     "variables:" yzuvpqrsl ; "weights:" 1, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1
     "ordering:" DEGREVLEX );
 NormalForm(1, Ideal(l(s − q), qr − s(p − l), uq − vp, s(u − l) − v(r − l),
                     (lv)z − 1));
  1
```

The theorem is not geometrically true. Let us look for some conditions.

```
 Elim(y..z, Ideal(l(s − q), qr − s(p − l), uq − vp, s(u − l) − v(r − l),
              (lv)z − 1));
  Ideal(vp − 1/2rs − 1/2sl, vr − 1/2rs − vl + 1/2sl, q − s, ps − rs − sl,
        pr − r^2 − pl + l^2, p^2 − r^2 − 2pl + l^2, up − 1/2r^2 − pl + 1/2l^2,
        vs − 1/2s^2, us − 1/2rs − 1/2sl, ur − 1/2r^2 − ul + 1/2l^2);
```

We choose the condition $ps - rs - sl$, which is nontrivial:

```
 NormalForm(1, Ideal(l(s − q), qr − s(p − l), uq − vp, s(u − l) − v(r − l),
                     (ps − rs − sl)z − 1));
  1
```

And we verify that the theorem is valid under this condition:

```
 NormalForm(1, Ideal(l(s − q), qr − s(p − l), uq − vp, s(u − l) − v(r − l),
                     (ps − rs − sl)z − 1, (vl)y − 1));
  0
```

What is the geometric meaning of this condition $s(p-r-l) \neq 0$? Well, $s \neq 0$ gives the perfectly reasonable condition that the parallelogram is non-degenerate (i.e., it is not a point), but $p - l \neq r$ gives the condition:

> the length of the projection of the segment ob onto oa is different from the length of the projection of the segment ac onto oa.

Obviously, this condition holds only for 'unusual' parallelograms. Our method requires further analysis, since we have proved a (false) 'theorem' that holds over an open set of the variety of parallelograms!

True Nondegeneracy Conditions

In order to bridge the gap between a geometry theorem and standard geometric intuition, we need a finer analysis of the conditions yielding 'degenerate cases'. The source of many problems is the fact that the hypotheses variety may have more components, some of which describe degenerate cases. To identify these, note that, when we first established the algebraic formulation of a geometry statement, we identified some variables as being independent. Now we emphasize this fact by calling such variables *geometrically independent*, because they correspond to coordinates of points in our geometric situation that can be chosen freely. (Such variables are not absolute: in choosing an algebraic formulation or even within such a formulation, different sets of variables may acquire this status; rescaling or applying translational or rotational invariance may reduce the number of geometrically independent variables.)

It seems natural to carry over the idea of geometrically independent variables to the hypotheses variety. Since we do not want to establish theorems that are only true in degenerate cases, finding nondegeneracy conditions should focus on exhibiting an open set of points in the hypotheses variety where the *geometrically independent variables* remain algebraically independent, i.e., such that *no polynomial in these variables vanishes over the open set.* To exploit this idea, we introduce the following concept of independence in the framework of commutative algebra.

Definition 4.14. Let I be an ideal of the polynomial ring $K[x_1,\dots,x_n]$. The variables $x_{i_1},\dots,x_{i_d} \in \{x_1,\dots,x_n\}$ are *independent modulo the ideal I* if $I \cap K[x_{i_1},\dots,x_{i_d}] = (0)$.

The *dimension* of an algebraic set $H \subset L^n$ and its ideal $I(H)$ is the number

$$\begin{aligned} d &= \dim(H) = \dim(I(H)) \\ &= \max\{r \mid \text{there are } r \text{ independent variables modulo } I(H)\}. \end{aligned}$$

In general, the dimension of an ideal coincides with the dimension of its radical; the special case that we need is proven in the following lemma.

Lemma 4.15. *Let I be an ideal of $K[x_1,\dots,x_n]$. Denote by $\mathbf{x}'$ the variables $(x_{i_1},\dots,x_{i_d})$. Then $I \cap K[\mathbf{x}'] = (0)$ if and only if $\sqrt{I} \cap K[\mathbf{x}'] = (0)$.*

Proof. The 'if' part follows immediately from $I \subset \sqrt{I}$. Conversely, if $I \cap K[\mathbf{x}'] = (0)$ and $g \in \sqrt{I} \cap K[\mathbf{x}']$, then $g^m \in I$ for some $m > 0$, so $g^m = 0$ and therefore $g = 0$.

Exercise 4.16.

1. Prove that a set $\mathbf{x}'$ of variables is independent modulo an ideal if and only if there is an isolated prime $\mathfrak{p}$ of this ideal such that $\mathbf{x}'$ is independent modulo $\mathfrak{p}$.

2. Prove that the dimension of an ideal agrees with the maximum dimension of its associated primes.

The following simple exercise shows that the concept of independent variables is quite tricky, at least when the ideal is not prime.

Exercise 4.17. Show that $\{x\}$ and $\{y, z\}$ are two maximal sets (with different cardinality) in $K[x, y, z]$ of independent variables modulo the ideal (xy, xz). Find the dimension of $\mathcal{Z}(xy, xz)$.

The connection between these algebraic results and the discussion at the beginning of this section is provided by the following easy result.

Proposition 4.18. *Let* x' *be a set of variables,* I *an ideal of* $K[x_1, \ldots, x_n]$. *The following statements are equivalent:*

(a) The set x' *is independent modulo* I.
(b) There is a nonempty open subset Γ *of some irreducible component of the variety* $\mathcal{Z}(I)$ *such that no nontrivial polynomial in the variables* x' *vanishes at every point of* Γ.
(c) There is a nonempty open subset Ω *of the variety* $\mathcal{Z}(I)$ *such that no nontrivial polynomial in the variables* x' *vanishes at every point of* Ω.

Proof. Assume x' is independent modulo I. Then x' is also independent modulo an isolated prime ideal of I. Since the irreducible components of $\mathcal{Z}(I)$ are the zeros of the isolated primes of I, it is enough to remark that a polynomial vanishes on an open set of an irreducible variety if and only if it vanishes on the whole variety. This yields *(a)* $\Rightarrow$ *(b)*. Now assume *(b)* holds. Then Γ is open (hence dense) in some irreducible component but is not, in general, open in $\mathcal{Z}(I)$. Since Γ is dense, it contains a nonempty open subset Ω of $\mathcal{Z}(I)$ (intersect Γ with the complement of the union of the remaining components). Hence *(b)* $\Rightarrow$ *(c)*. The implication *(c)* $\Rightarrow$ *(a)* is trivial.

Remark 4.19. Besides the brute force way of using the CoCoA command `Elim` to check if a set of variables is independent and finding a maximum set of independent variables, the command `Dim` provides a direct way to find the dimension.

Of course, the variables we choose as geometrically independent in the formulation of a geometry theorem should be independent variables of $K[\mathrm{x}, \mathrm{y}]$ modulo the hypotheses ideal $I(H)$ in order to guarantee that at least we have an open set of nondegenerate cases in the hypotheses variety. Such an open set meets (and is dense in) some irreducible component of H, but misses entirely components describing degenerate cases. So we look for conditions among polynomials in these variables. This motivates the following definition and proposition.

Definition 4.20. A nonzero polynomial $g \in K[\mathrm{x}, \mathrm{y}]$ is a *true nondegeneracy condition* for a geometry theorem $\mathcal{T}$ if $g \in K[\mathrm{x}]$, where $\mathrm{x} = (x_1, \dots, x_d)$ is a set of geometrically independent variables over the hypotheses variety, that is, modulo its ideal, and $\mathcal{T}$ is geometrically true under the condition $g \neq 0$.

Notice that such conditions are always nontrivial, since they do not belong to the ideal of the hypotheses variety (by definition of independent variables), hence (by Lemma 4.15) not to its radical.

Proposition 4.21. *Retain the notation of Definition* 4.20. *The following statements regarding the geometry theorem* $\mathcal{T}$ *are equivalent.*

(a) There is a true nondegeneracy condition for $\mathcal{T}$.
(b) There is $g \in K[\mathbf{x}] \setminus \{0\}$ *such that* $g \cdot t \in I(H) = \sqrt{(h_1, \dots, h_p)}$.
(c) $I_c = (h_1, \dots, h_p, tz - 1) \cap K[\mathbf{x}] \neq (0)$.
(d) t *vanishes on every irreducible component of* H *where* $\mathbf{x}$ *is a set of geometrically independent variables.*

If one (hence all) of them holds, we say that the theorem $\mathcal{T}$ is *generically true.*

Proof. We leave it to the reader to show that *(a)*, *(b)* and *(c)* are equivalent.

Now let us assume *(b)* and let H_i be an irreducible component of H where $\mathbf{x}$ is an independent set of variables. If $g \in K[\mathbf{x}] \setminus \{0\}$ is such that $g \cdot t \in \sqrt{(h_1, \dots, h_p)} = I(H)$, then $g \cdot t \in I(H_i)$. Now, $g \notin I(H_i)$, because $\{x_1, \dots, x_d\}$ is independent modulo $I(H_i)$. Since $I(H_i)$ is a prime ideal, it follows that $t \in I(H_i)$.

Conversely, let $H = H_1 \cup \cdots \cup H_r \cup H_1^* \cup \cdots \cup H_l^*$ be the decomposition of H in irreducible components, labeled so that $\mathbf{x}$ is a set of independent variables precisely over each H_i. As $\{x_1, \dots, x_d\}$ is dependent modulo $I(H_j^*)$, for each $j = 1, \dots, l$, there is a nonzero $g_j \in K[\mathbf{x}]$ such that g_j vanishes on H_j^*. Take $g = g_1 \cdots g_l$ (if $l = 0$ choose $g = 1$); then $g \cdot t$ vanishes on H.

Remark 4.22. By Remark 4.8, nondegeneracy conditions for a statement $\mathcal{T}$ are to be found in the elimination ideal (or, rather, in its radical, but we follow here the same simplification as in 4.3) $I_c = (h_1, \dots, h_p, tz - 1) \cap K[x_1, \dots, x_d]$. Using CoCoA we obtain a Gröbner basis of I_c by

$$\mathsf{Elim}(y_1..y_r, \mathsf{Ideal}(h_1, \dots, h_p, tz - 1))$$

We distinguish two cases:

1) Following Proposition 4.21, we say that theorem $\mathcal{T}$ is *not generically true* if $I_c = (0)$. In terms of algebraic geometry this means that $\{t \neq 0\}$ holds over some 'geometrically relevant' component of H. In most cases it also means that $\{t \neq 0\} \cap H$ has the same dimension as H (since degenerate components 'should' have smaller dimension), but see Exercise 4.9.

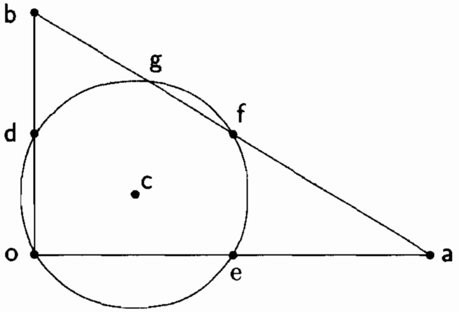
2) If $I_c = (g_1, \ldots, g_s) \neq (0)$, then

$$h_1 = 0, \ldots, h_p = 0 \text{ and } (g_1 \neq 0 \text{ or } g_2 \neq 0 \text{ or } \ldots \text{ or } g_s \neq 0) \Rightarrow t = 0.$$

We leave to the reader the task of finding the geometrical interpretation of the zeroset $\mathcal{Z}(g_1, \ldots, g_s)$.

This analysis implies that Example 4.13 is not generically true since there are no conditions in the independent variables l, r, s.

Example 4.23. In any right triangle the circle passing through the midpoints of the sides also contains the feet of the three altitudes.



Consider the triangle with vertices $\mathsf{o} = (0,0)$, $\mathsf{a} = (2r, 0)$ and $\mathsf{b} = (0, 2s)$. Let $\mathsf{d} = (0, s)$, $\mathsf{e} = (r, 0)$ and $\mathsf{f} = (r, s)$. Denote by $\mathsf{c} = (p, q)$ the center of the circle passing through the points d, e, and f. Let $\mathsf{g} = (u, v)$ be the foot of the altitude from o. Remark that r, s are the geometrically independent variables.

Hypotheses: $|\mathsf{cd}| = |\mathsf{ce}|$: $h_1 = (r-p)^2 + q^2 - p^2 - (q-s)^2 = 0$
$|\mathsf{cd}| = |\mathsf{cf}|$: $h_2 = (r-p)^2 + (s-q)^2 - p^2 - (q-s)^2 = 0$
$\mathsf{g} \in \mathsf{ab}$: $h_3 = r(v-2s) + su = 0$
$\mathsf{og} \perp \mathsf{ab}$: $h_4 = ru - sv = 0$

Thesis: $|\mathsf{cd}| = |\mathsf{cg}|$: $t = (u-p)^2 + (v-q)^2 - p^2 - (q-s)^2 = 0$

```
Ring ( "ring name:" R ; "characteristic:" 0 ;
    "variables:" zuvpqrs ; "weights:" 1, 1 , 1 , 1 , 1 , 1 , 1
    "ordering:" DEGREVLEX );
```

$$\mathsf{NormalForm}(1, \mathsf{Ideal}((r-p)^2 + q^2 - p^2 - (q-s)^2, r(v-2s) + su,$$
$$(r-p)^2 + (s-q)^2 - p^2 - (q-s)^2, ru - sv,$$
$$((u-p)^2 + (v-q)^2 - p^2 - (q-s)^2)z - 1));$$

1

The theorem is not geometrically true. Thus, we look for true nondegeneracy conditions:

```
Elim(z..q, Ideal((r − p)^2 + q^2 − p^2 − (q − s)^2,
        (r − p)^2 + (s − q)^2 − p^2 − (q − s)^2, r(v − 2s) + su, ru − sv,
        ((u − p)^2 + (v − q)^2 − p^2 − (q − s)^2)z − 1));
  Ideal(s, r);
```

Therefore this theorem is generically true. It fails only for degenerate triangles, i.e., when $s = r = 0$.

Exercise 4.24 (Simson's Theorem). The pedal points (feet) of the perpendiculars drawn from an arbitrary point on a triangle's circumscribed circle to the three sides are collinear.

1. Let C be the circumcircle with center $\mathsf{c} = (p, q)$ of the triangle with vertices $\mathsf{o} = (0,0)$, $\mathsf{a} = (l, 0)$ and $\mathsf{b} = (r, s)$. Set up equations describing hypotheses and thesis for the theorem.
2. Show that Simson's Theorem is generically true and derive a true nondegeneracy condition for its validity. Phrase this as a condition on the sides of the triangle.

5. Searching for Extra Hypotheses

So far our method identifies a theorem's validity in nondegenerate cases. It discovers, essentially, statements that hold over open sets of the hypotheses variety. But, unless one is very lucky (or clever), most properties that one states 'at random' about a certain geometric setting will not be generally true. For instance, a statement that is false for general triangles, may hold for special kinds of triangle. For these theorems that are not generically true, our method has nothing to say (except that they are not true). Our next task is to find, if possible, extra hypotheses so that the resulting statement will be generically true over the new set of hypotheses (see [4] for a detailed account of this method). A natural ideal to study in this context is the ideal $(h_1, \ldots, h_p, t)$, because it describes the set where both the hypotheses and the thesis hold. Since we should look for new hypotheses that are expressible in terms of the independent geometric variables, we consider $(h_1, \ldots, h_p, t) \cap K[\mathbf{x}]$, where $\mathbf{x} = (x_1, \ldots, x_d)$ is a distinguished set of geometrically independent variables on the hypotheses variety. The proof of the following statement is omitted, since it is very similar to that of Proposition 4.21.

Proposition 5.1. *The following statements are equivalent.*

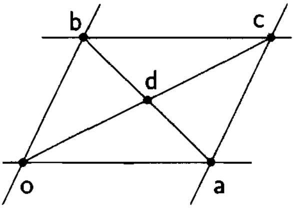
a) $(h_1, \ldots, h_p, t) \cap K[\mathbf{x}] \neq (0)$.

b) ***t vanishes on none of the irreducible components of the hypotheses variety H where the variables $\mathbf{x}$ are independent.***

In this case we say that the theorem is *generically false*.

If a given thesis t is generically false under the hypotheses $h_1, \dots, h_p$, we consider the nontrivial ideal $(h_1, \dots, h_p, t) \cap K[\mathrm{x}] \neq (0)$; it is not contained in the radical of the hypotheses ideal, since $\sqrt{(h_1, \dots, h_p)} \cap K[\mathrm{x}] = (0)$. Adding a nontrivial $h \in (h_1, \dots, h_p, t) \cap K[\mathrm{x}]$ to our hypotheses, yields a strictly larger hypotheses ideal and therefore a strictly smaller hypotheses variety; it may also affect the set of independent variables. The new hypotheses variety must now be analyzed via the standard procedure, searching for nondegeneracy conditions and so on. No guarantee that the new collection of hypotheses will yield a generically true theorem, but we can try...!

Example 5.2. In any parallelogram, the diagonals intersect at a right angle.



Consider a parallelogram as in Example 4.13, with vertices $\mathsf{o} = (0,0)$, $\mathsf{a} = (l,0)$, $\mathsf{b} = (r,s)$ and $\mathsf{c} = (p,q)$. The independent variables are r, s, l.

Hypotheses: $\mathsf{oa} \parallel \mathsf{bc}$: $h_1 := l(s-q) = 0$
$\mathsf{ob} \parallel \mathsf{ac}$: $h_2 := qr - s(p-l) = 0$

Thesis: $\mathsf{oc} \perp \mathsf{ab}$: $t := p(r-l) + qs = 0$

Ring ("ring name:" R ; "characteristic:" 0 ;
"variables:" zpqrsl ; "weights:" 1, 1 , 1 , 1 , 1 , 1
"ordering:" DEGREVLEX);

NormalForm$(1, \mathsf{Ideal}(l(s-q), qr - s(p-l), (p(r-l)+qs)z - 1))$;
1

Elim$(z..q, \mathsf{Ideal}(l(s-q), rq - s(p-l), (p(r-l)+qs)z - 1))$;
Ideal(0);

Elim$(z..q, \mathsf{Ideal}(l(s-q), rq - s(p-l), p(r-l)+qs))$;
Ideal$(r^2sl + s^3l - sl^3)$;

NormalForm$(1, \mathsf{Ideal}(l(s-q), rq - s(p-l), r^2sl + s^3l - sl^3,$
$(p(r-l)+qs)z - 1)))$;
1

Elim$(z..r, \mathsf{Ideal}(l(s-q), rq - s(p-l), r^2sl + s^3l - sl^3,$
$(p(r-l)+qs)z - 1)))$;
Ideal(sl);

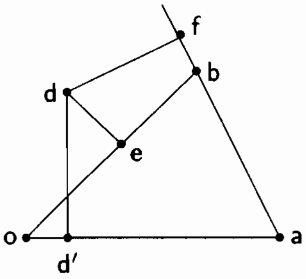
So the theorem is generically false, but the third computation shows a new hypothesis,

$$g = r^2sl + s^3l - sl^3 = sl(r^2 + s^2 - l^2),$$

that makes the theorem generically true (remark that adding this hypothesis reduces the set of independent variables to s, l). More specifically, we discover that *the theorem is true if the sides of the parallelogram are equal, namely, if* $r^2 + s^2 = l^2$ *(i.e., when it is a square or a rhomboid) and if the parallelogram does not collapse:* $sl \neq 0$.

The next example shows how to discover the converse of Simson's theorem.

Example 5.3. Consider a triangle and assume, without loss of generality, that the vertices have coordinates $\mathsf{o} = (0,0)$, $\mathsf{a} = (l,0)$, $\mathsf{b} = (r,s)$; let $\mathsf{d} = (m,n)$ be an arbitrary point in the plane. We assign coordinates to the feet of the perpendiculars dropped from d to the three sides of the triangle: $\mathrm{e} = (v,w)$, $\mathsf{f} = (t,u)$, $\mathsf{d}' = (m,0)$. We conjecture that these three points are collinear.



This construction yields the following equations:

Hypotheses:

$$\begin{array}{ll} \mathsf{e} \in \mathsf{ob}: & sv - rw = 0 \\ \mathsf{ob} \perp \mathsf{de}: & r(m-v) + s(n-w) = 0 \\ \mathsf{f} \in \mathsf{ab}: & s(t-l) - u(r-l) = 0 \\ \mathsf{ab} \perp \mathsf{df}: & (t-m)(r-l) + s(u-n) = 0 \end{array}$$

Next we conjecture, in this situation, that points $\mathsf{e}, \mathsf{f}, \mathsf{d}'$ are collinear (perhaps because they look like lying on a line in the above figure); i.e., $(w-u)(m-t) + u(v-t) = 0$. Not surprisingly, it turns out that

$$\begin{aligned} \mathsf{NormalForm}(1, \mathsf{Ideal}(&s(t-l) - u(r-l), (t-m)(r-l) + s(u-n), sv - rw, \\ &r(m-v) + s(n-w), z(w(t-m) - u(v-m)) - 1)) = 1 \end{aligned}$$

so the conjecture is not geometrically true. But it also happens that elimination of the slack variable z from the ideal yields an ideal not contained in the radical of the hypotheses ideal. So the conjecture holds over an open set of the hypotheses variety! Some extra computations confirm that this open set lies entirely in a degenerate locus of the hypotheses variety (in fact, it is contained in the subset where $s = 0$). This is possible, as remarked above, because this hypotheses variety has components of dimension 6, while there are only 5 independent variables (m, n, r, s, l) from a geometric point of view. On the other hand, if we eliminate the slack variable z plus the geometrically dependent variables v, w, t, u, we get the zero ideal, so the conjecture is not generically true over an open set of nondegenerate cases:

$$\begin{aligned}&\mathsf{Elim}(z..u, \mathsf{Ideal}(s(t-l)-u(r-l), (t-m)(r-l)+s(u-n), sv-rw,\\&\qquad r(m-v)+s(n-w), z(w(t-m)-u(v-m))-1));\\&\mathsf{Ideal}(0);\end{aligned}$$

Now we start again, this time eliminating all the geometrically dependent variables, i.e., the variables v to u in the set $\{v, w, t, u, n, m, r, s, l\}$, from the ideal generated by the hypotheses plus the thesis:

$$\begin{aligned}&\mathsf{Elim}(v..u, \mathsf{Ideal}(s(t-l)-u(r-l), (t-m)(r-l)+s(u-n), sv-rw,\\&\qquad r(m-v)+s(n-w), ((w-u)(m-t)+u(v-t))));\\&\mathsf{Ideal}(nr^2s^2l-m^2s^3l-n^2s^3l+ns^4l-nrs^2l^2+ms^3l^2);\end{aligned}$$

This yields an extra hypothesis:

$$nr^2s^2l-m^2s^3l-n^2s^3l+ns^4l-nrs^2l^2+ms^3l^2=0.$$

Now we observe that sl is a common factor, and its vanishing clearly corresponds to degenerate cases of 'flat' triangles. After removing this factor, the equation $nr^2s-m^2s^2-n^2s^2+ns^3-nrsl+ms^2l=0$ remains. Since for a given triangle the values of l, r, s will be fixed, the above equation should be regarded as one in the variables m, n. Then it is the equation of a circle, passing through the three vertices of the triangle. Thus our conjectural statement is not true in general, but it *could be true either if the triangle degenerates or the given point d is not arbitrary, but lies on the circle determined by the vertices of the triangle.* Over nondegenerate triangles the last condition is therefore necessary. It is easy to check that this condition is also sufficient (with some nondegeneracy conditions). Indeed, as explained above, we add one extra variable z, and proceed to eliminate, in the ideal generated by all the hypotheses (old ones plus the newly discovered) and the thesis (multiplied by z and subtracting 1), all non-independent variables from $\{z, v, w, t, u, n, m, r, s, l\}$:

$$\begin{aligned}&\mathsf{Elim}(z..n, \mathsf{Ideal}(s(t-l)-u(r-l), (t-m)(r-l)+s(u-n), sv-rw,\\&\qquad r(m-v)+s(n-w), nr^2s-m^2s^2-n^2s^2+ns^3-nrsl+ms^2l,\\&\qquad ((w-u)(m-t)+u(v-t))z-1))\\&\mathsf{Ideal}(r^4+2r^2s^2+s^4-2r^3l-2rs^2l+r^2l^2+s^2l^2);\end{aligned}$$

Now this nondegeneracy condition is $(r^2+s^2)((r-l)^2+s^2) \neq 0$, which expresses – over the reals – that the vertices of the triangle should not coincide. Thus we have, so to speak, rediscovered Simson's Theorem starting from a wrong assumption.

We finish with a couple of exercises on this technique of automatic discovery of theorems.

Exercise 5.4. In a triangle with vertices $\mathsf{a} = (b,0)$, $\mathsf{b} = (0,a)$, $\mathsf{c} = (1,0)$, consider a point $\mathsf{d} = (\mathrm{c}, d)$ on the line ab, and the following lengths: the distance from d to ac $(= x)$, the distance from d to bc $(= y)$ and the length of the altitude from b to the opposite side $(= z)$. Then, the algebraic sum of any two of these lengths is equal to the third one.

1. Denote by $\mathsf{e} = (u,v)$ the intersection point of bc with its perpendicular from d and let $\mathsf{f} = (c,0)$. Set up equations describing hypotheses and thesis (you must assign some signs to the lengths according to the position of d in ab).
2. Show that the theorem is generically false; add a new hypothesis. Can you describe its meaning?

Exercise 5.5. In a triangle, the orthocenter (intersection of altitudes), the centroid (intersection of medians), the circumcenter (center of the circle circumscribed about the triangle) and the incenter (center of the circle inscribed in the triangle) lie on a line (the *Euler line*).

1. Consider the triangle with vertices $\mathsf{a} = (-1,0)$, $\mathsf{b} = (1,0)$, $\mathsf{c} = (a,b)$. Show that the following statement is generically true: the orthocenter $\mathsf{d} = (p,q)$, the circumcenter $\mathsf{e} = (u,v)$ and the centroid $\mathsf{f} = (l,r)$ lie on a line.
2. Next, we investigate the statement: the incenter $\mathsf{g} = (s,w)$, the circumcenter $\mathsf{e} = (u,v)$ and the centroid $\mathsf{f} = (l,r)$ lie on a line. The incenter $\mathsf{g} = (s,w)$ is the center of the circle of radius w: $(x-s)^2+(y-w)^2-w^2 = 0$ that is tangent to the sides of the triangle. Find the equations of this point by eliminating the variables x, y from the equations of the circle with center (s,w) and radius w and from the equations giving the perpendicularity from a radius of the circle to ac (respectively bc): $b(x+1)-(a+1)y$ (respectively $b(x-1)-(a-1)y$).
3. Is the new theorem generically true or generically false?
4. Introduce an extra hypothesis expressing that the triangle be isosceles. Is the new theorem generically true or generically false?

References

1. S.C. Chou (1987): *Mechanical Geometry Theorem Proving*, D. Reidel.
2. D. Cox, J. Little, and D. O'Shea (1992): *Ideals, Varieties, and Algorithms*, Undergraduate Texts in Mathematics, Springer-Verlag, New York Berlin Heidelberg.

3. D. Kapur (1986): *Geometry theorem proving using Hilbert's Nullstellensatz*, pp. 202–208, in: Proc. of the 1986 Symposium on Symbolic and Algebraic Computation, Ed. B.W. Char, ACM Press, Waterloo.
4. T. Recio and M. P. Vélez (1996): *Automatic discovery of theorems in elementary geometry*, submitted to Journal of Automated Reasoning.
5. W. T. Wu (1994): *Mechanical Theorem Proving in Geometries*, Texts and Monographs in Symbolic Computation, Springer-Verlag, Berlin Heidelberg New York.

Project 2. The Birkhoff Interpolation Problem

Maria-Jose Gonzalez-Lopez and Laureano Gonzalez-Vega

1. Introduction

The problem of interpolating an unknown function $f\colon \mathbb{R} \to \mathbb{R}$ by a univariate polynomial with knowledge of the values of f and some of its derivatives at some points in $\mathbb{R}$ is one of the main problems in Numerical Analysis and Approximation Theory.

Two classical interpolation cases have been widely studied and solved: the Lagrange Interpolation Formula and the Hermite Interpolation Problem. In the first case the values of f at the points $x_1 < \cdots < x_n$ are known and the Lagrange Interpolation Formula shows that there exists a unique polynomial of degree less than or equal to $n-1$ with the same behaviour as f at the points x_i.

The Hermite Interpolation Problem generalizes the previous case by including some information coming from the derivatives of f. Let $x_1 < \cdots < x_n$ be given points and $\nu_1, \ldots, \nu_n$ positive integers; the Hermite Interpolation Problem is solved by proving that there exists a unique polynomial P (which is explicitly given) of degree less than or equal to

$$N = \nu_1 + \cdots + \nu_n - 1$$

such that for every $k \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, \nu_k - 1\}$ the following equality is satisfied, where $f^{(j)}$ denotes the j-th derivative of f:

$$P^{(j)}(x_k) = f^{(j)}(x_k).$$

The main purpose of this project is to show how some of the recipes introduced in Chapter 6 can be used in order to determine which interpolation schemes (more general than Lagrange or Hermite) are good in the sense that for any choice of nodes and function values there is one and only one solution.

2. Poised Matrices

The problem of interpolation by polynomials can be presented in a general way by describing the interpolation conditions in terms of incidence matrices: such matrices will contain the information known about f.

Definition 2.1. Let n and r be two integers such that $n \geq 1$ and $r \geq 0$. The $n \times (r+1)$ matrix

$$E = \begin{pmatrix} e_{1,0} & \cdots & e_{1,r} \\ \vdots & & \vdots \\ e_{n,0} & \cdots & e_{n,r} \end{pmatrix}$$

is called an *incidence matrix* if $e_{i,j} \in \{0,1\}$ for every i and j.

For an incidence matrix E, the symbol $|E|$ will denote the number of ones in E:

$$|E| = \sum_{i,j} e_{i,j}.$$

In the case where $|E|$ is equal to the number of columns in E, the incidence matrix E is called *normal*.

Let E be an incidence matrix of dimension $n \times (r+1)$, $X = \{x_1, \ldots, x_n\}$ a set of real numbers such that $x_1 < \cdots < x_n$, and F a matrix of given real numbers with the same dimensions as E, whose entries are denoted by $f_{i,j}$. The *Birkhoff Interpolation Problem* consists of determining a polynomial $P \in \mathbb{R}[x]$ of degree smaller than or equal to r which interpolates F at (X, E), i.e., which satisfies the conditions:

$$P^{(j)}(x_i) = f_{i,j}, \tag{2.1}$$

where the indices (i,j) are those for which $e_{i,j} = 1$.

Definition 2.2. An incidence matrix E with n rows and $r+1$ columns is said to be *poised* if, for each choice of the nodes $x_1 < \cdots < x_n$ and matrix F, there exists a unique polynomial $P \in \mathbb{R}[x]$ of degree smaller than or equal to r which interpolates F at (X, E).

Example 2.3. The incidence matrix corresponding to the Lagrange Interpolation Formula,

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix},$$

is poised.

A second example comes from the Hermite Interpolation Problem:

Exercise 2.4. Describe the corresponding poised incidence $n \times N$-matrix for any choice of positive integers $\nu_1, \ldots, \nu_n$, with $N = \nu_1 + \cdots + \nu_n - 1$.

The complete characterization of poised matrices is still an open problem. In the literature on numerical analysis and approximation theory one can find several sufficient conditions for an incidence matrix to be poised, but a complete characterization has not been found, not even for the cases where the number of nodes is small, for example 3 or 4 (see for example [1]).

In order to use the techniques introduced in Chapter 6 for determining all poised matrices with fixed dimension, first:

Exercise 2.5. Prove that the only incidence matrices which are poised are those which are normal.

The next exercise presents a complete solution for the case $n = 2$.

Exercise 2.6. A characterization of all poised matrices for the case $n = 2$ and arbitrary r is given by the so-called Pólya Condition:

> A normal incidence matrix E with n rows and $r + 1$ columns satisfies the Pólya Condition if, for every k in $\{0, \dots, r\}$, the following inequality holds:
> $$\sum_{i=1}^{n} \sum_{j=0}^{k} e_{i,j} \geq k + 1.$$

For a normal incidence matrix E with 2 rows and $r + 1$ columns give a proof of the equivalence

$$E \text{ is poised} \iff E \text{ satisfies the Pólya Condition.}$$

Hint: use Rolle's Theorem (see [1] or [4]).

To deal with the general case, we consider the matrix M_E associated with the linear system of equations in (2.1), giving the interpolating polynomial P for X and E, and its determinant D_E. Thus, the problem of determining the non-poisedness of a normal incidence matrix E is reduced to finding a set $X = \{x_1, \dots, x_n\}$ of real numbers satisfying

$$x_1 < \cdots < x_n \qquad \text{and} \qquad D_E(x_1, \dots, x_n) = 0.$$

Since every polynomial D_E is usually divisible by powers of several $(x_i - x_j)$, the polynomial resulting from division by all these factors is denoted by $\widetilde{D}_E$ and called the *poise-indicator* of E. The main property of $\widetilde{D}_E$ related to the poisedness of E is shown in the following theorem and corollary, whose proofs can be found in [2] and are guided in the following exercises.

Theorem 2.7. *Let E be a normal incidence matrix with n rows and $r + 1$ columns. Then, if $t_1, \dots, t_{n-1}$ are new variables, the polynomial:*

$$H_E = \widetilde{D}_E(x_1, x_1 + t_1^2, x_1 + t_1^2 + t_2^2, \dots, x_1 + \sum_{i=1}^{n-1} t_i^2)$$

is a homogeneous polynomial in $\mathbf{Z}[t_1, \dots, t_{n-1}]$.

Exercise 2.8. Guided proof of Theorem 2.7 (see [2]):

a) Prove that H_E belongs to $\mathbb{Z}[t_1, \ldots, t_{n-1}] = \mathbb{Z}[\underline{t}]$.

Suppose that $H_E = H_E(\underline{t}, x_1)$ has degree $d > 0$ in x_1. Then there exists $\underline{t}^* = (t_1^*, \ldots, t_{n-1}^*) \in \mathbb{C}^{n-1}$ such that $H_E(\underline{t}^*, x_1)$ is a univariate polynomial of degree $d > 0$. Let $\alpha^* \in \mathbb{C}$ be such that $H_E(\underline{t}^*, \alpha^*) = 0$. Prove that this implies the contradiction: $H_E(\underline{t}^*, \alpha^* + \beta) = 0$ for all $\beta \in \mathbb{R}$.

Hint: suppose that there exists $\beta^* \in \mathbb{R}$ such that $H_E(\underline{t}^*, \alpha^* + \beta^*) \neq 0$; relate the two (unique) Birkhoff interpolating polynomials for E and the nodes

$$\left\{\alpha^* + \beta^*, \alpha^* + \beta^* + t_1^{*2}, \ldots, \alpha^* + \beta^* + \sum_{i=1}^{n-1} t_i^{*2}\right\},$$

and for E and the nodes

$$\left\{\alpha^*, \alpha^* + t_1^{*2}, \ldots, \alpha^* + \sum_{i=1}^{n-1} t_i^{*2}\right\}$$

to obtain a contradiction.

b) Prove that $H_E(\underline{t})$ is homogeneous.

Let λ be a new variable, and consider the polynomial

$$R(\underline{t}, \lambda) := H_E(\lambda t_1, \ldots, \lambda t_{n-1}) = \sum_{j=0}^{r} b_j(\underline{t})\lambda^j.$$

Prove that there is at most one $b_j(\underline{t})$ nonzero: if two of them were different from zero then there exist $t^* \in \mathbb{R}^{n-1}$ such that $H_E(t^*) \neq 0$, and $\lambda^* \in \mathbb{C}$ such that $R(t^*, \lambda^*) = 0$. Relate the two (unique) Birkhoff interpolating polynomials for E and the nodes

$$\left\{0, t_1^{*2}, t_1^{*2} + t_2^{*2} \ldots, \sum_{i=1}^{n-1} t_i^{*2}\right\},$$

and for E and the nodes

$$\left\{0, \lambda^{*2} t_1^{*2}, \lambda^{*2}(t_1^{*2} + t_2^{*2}) \ldots, \lambda^{*2} \sum_{i=1}^{n-1} t_i^{*2}\right\}$$

to obtain a contradiction. Conclude that there exists an m such that:

$$H_E(\lambda t_1, \ldots, \lambda t_{n-1}) = \lambda^m H_E(\underline{t}).$$

Corollary 2.9. *Let E be a normal incidence matrix. Then E is not poised if and only if the polynomial H_E has a real solution* $(1, t_2, \ldots, t_{n-1})$ *such that* $t_2 \cdots t_{n-1} \neq 0$.

Exercise 2.10. **Prove the previous corollary.**

3. Examples

This section is devoted to showing how to use the previous theorem and corollary to determine if a normal incidence matrix is poised or not. The main computational tools to use have been presented in Chapter 6: the Sturm-Habicht and Sylvester-Habicht sequences and the Sign Determination Scheme (`Recipe SI`).

Example 3.1. We present a particular case, showing all the objects presented in the previous section related to the normal incidence matrix E defined by:

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

The matrix M_E associated to E is

$$M_E = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 0 & 1 & 2x_2 & 3x_2^2 & 4x_2^3 & 5x_2^4 \\ 0 & 1 & 2x_3 & 3x_3^2 & 4x_3^3 & 5x_3^4 \\ 0 & 0 & 2 & 6x_2 & 12x_2^2 & 20x_2^3 \\ 0 & 0 & 0 & 6 & 24x_1 & 60x_1^2 \end{pmatrix}.$$

The corresponding polynomial D_E factorizes in the following way:

$$D_E = -36(x_2 - x_3)^2(x_1 - x_2)^4(6x_1^2 - 12x_3x_1 - x_2^2 + 2x_2x_3 + 5x_3^2).$$

Thus the poise-indicator of E is:

$$\widetilde{D}_E = -36(6x_1^2 - 12x_3x_1 - x_2^2 + 2x_2x_3 + 5x_3^2),$$

and the polynomial H_E of Theorem 2.7 is

$$H_E(t_1, t_2) = -36(5t_1^4 + 12t_1^2t_2^2 + 6t_2^4).$$

This allows us to conclude that E is poised since for every $x_1 < x_2 < x_3$ the polynomial $\widetilde{D}_E$ is strictly negative:

$$x_1 < x_2 < x_3 \implies \widetilde{D}_E < 0 \implies D_E < 0.$$

Given n and r, the computation of all poised matrices for n nodes and degree r has been reduced to determining, for a normal incidence matrix E, whether the polynomial $H_E(1, t_2, \dots, t_{n-1})$ has no solutions with nonzero coordinates. This is a problem that can be classified as a Quantifier Elimination Problem. What is required for every E, is whether or not the following assertion holds:

$$\exists t_2 \in \mathbb{R}, \ \cdots, \ \exists t_{n-1} \in \mathbb{R} \quad H_E(1, t_2, \dots, t_{n-1}) = 0 \quad \text{and} \quad t_2 \cdots t_{n-1} \neq 0.$$

Example 3.2. Let E be the following normal incidence matrix

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

whose poisedness we want to determine. In this case, the polynomial to be considered is the following one:

$$H_E(1, t_2, t_3) = 1 + 4t_2^2 + 6t_2^4 - 6t_3^4 + 2t_2^6 - 6t_2^2 t_3^4 - 4t_3^6.$$

The principal Sturm-Habicht coefficients of $H_E(1, t_2, t_3)$ (i.e., the principal Sylvester-Habicht coefficients of $H_E(1, t_2, t_3)$ and $\partial H_E(1, t_2, t_3)/\partial t_3$) with respect to t_3 are:

$$\begin{aligned}
S_6 &= -1, \\
S_5 &= -1, \\
S_4 &= 1 + t_2^2, \\
S_3 &= 1 + 3t_2^2 + 3t_2^4 + t_2^6, \\
S_2 &= 2t_2^{12} + 12t_2^{10} + 28t_2^8 + 33t_2^6 + 21t_2^4 + 7t_2^2 + 1, \\
S_1 &= 8t_2^{16} + 60t_2^{14} + 180t_2^{12} + 284t_2^{10} + 264t_2^8 + 152t_2^6 + 54t_2^4 + 11t_2^2 + 1, \\
S_0 &= -32t_2^{22} - 320t_2^{20} - 1352t_2^{18} - 3192t_2^{16} - 4728t_2^{14} - 4692t_2^{12} - 3232t_2^{10} \\
&\quad -1564t_2^8 - 526t_2^6 - 118t_2^4 - 16t_2^2 - 1,
\end{aligned}$$

which, in this particular case, have constant sign for every $t_2 \in \mathbb{R}$:

$$S_6 < 0,\ S_5 < 0,\ S_4 > 0,\ S_3 > 0,\ S_2 > 0,\ S_1 > 0,\ S_0 < 0.$$

When the principal Sturm-Habicht coefficients do not vanish, the formula giving the number of real roots of $H_E(1, \alpha, t_3)$ for any $\alpha \in \mathbb{R}$ is equal to the difference between the permanences and variations of sign changes in $\{-, -, +, +, +, +, -\}$ (`Recipe CRS`$_1$): for any $\alpha \in \mathbb{R}$ the polynomial $H_E(1, \alpha, t_3)$ has two different real roots; this allows us to conclude that E is not poised.

The situation where the signs of the principal Sturm-Habicht coefficients are constant appears quite often but, in general, this is not the case, and so, in order to determine the poisedness of a normal incidence matrix, we are faced with the problem of determining the sign conditions satisfied by a list of univariate polynomials (`Recipe SI` of Chapter 6). In the following exercise we present a guided example of this fact.

Exercise 3.3. Consider the following normal incidence matrix:

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

whose poise-indicator is (once t_1 is replaced by 1):

$$H_E(1, t_2, t_3) = 8 + 6t_2^2 + 6t_3^2 - t_2^6 - 3t_2^4t_3^2 - 3t_2^2t_3^4 - t_3^6.$$

1. Compute the (six) principal Sturm-Habicht coefficients S_i, $i = 1, \ldots, 6$, of $H_E(1, t_2, t_3)$ with respect to t_3.
2. Deduce that, for $t_2 \neq 0$, the only interesting sign conditions are those satisfied by the polynomials S_1 and S_0.
3. Using `Recipe SI` (which allows one to compute easily the nonempty sign conditions satisfied by the real roots of an univariate polynomial or a finite list of univariate polynomials), compute the nonempty sign conditions satisfied by the polynomials S_1 and S_0.
   Hint: It suffices to look first at the signs at infinity of S_1 and S_0 and then apply `Recipe SI` to the polynomial $(S_1 \cdot S_0)'$ with the list $\{S_1, S_0\}$. Besides, recall that we are only interested in those sign conditions with $t_2 \neq 0$.
4. Conclude that E is not poised.

4. Conclusions

A mixture of some basic algorithms used to deal with Quantifier Elimination Problems allow to determine all the normal incidence matrices which are poised for the following cases (see [2]):

- $n = 2$ (and any degree; see the Pólya Condition above),
- $r = 3$ (and any number of nodes),
- $r = 4$ and $n = 3$ or 4,
- $r = 5$ and $n = 3$.

For example, to determine the 6454 poised matrices for degree 5 and 3 nodes, a check of all 18564 normal incidence matrices was needed. These numbers give an idea of the amount of computation that was necessary to solve the Birkhoff Interpolation Problem for these particular cases.

Another consequence of this computation was the generation of results on the number of poised matrices for an arbitrary number of nodes and fixed degree. In this context, if $m_{n,r}$ denotes the number of poised matrices for degree r and n nodes, then the proof of the following equalities can be found in [2]:

$$m_{n,1} = \binom{n}{1} + 3\binom{n}{2}$$

$$m_{n,2} = \binom{n}{1} + 12\binom{n}{2} + 15\binom{n}{3}$$

$$m_{n,3} = \binom{n}{1} + 40\binom{n}{2} + 135\binom{n}{3} + 102\binom{n}{4}.$$

One of the main applications of the Birkhoff Interpolation Schemes appears when reliably controlling the defect in the numerical solution of nonstiff initial value problems by using Runge-Kutta schemes of any order (see for example [3]).

References

1. B.D. Bojanov, H.A. Hakopian, and A.A. Sahakian (1993): *Spline Functions and Multivariate Interpolations.* Series Mathematics and Its Applications **248**, Kluwer Academic Publishers.
2. L. Gonzalez-Vega (1996): *Applying quantifier elimination to the Birkhoff interpolation problem.* Journal of Symbolic Computation **22**, 83–103.
3. D.J. Higham (1991): *Runge-Kutta defect control using Hermite-Birkhoff interpolation.* SIAM J. Sci. Stat. Comput. **12**, 5, 991–999.
4. B. Sendov and A. Andreev (1994): *Aproximation and interpolation theory.* Handbook of Numerical Analysis, Volume III, eds. P.G. Ciarlet and J.L. Lions, 229–300, North-Holland.

Project 3. The Inverse Kinematics Problem in Robotics

Maria-Jose Gonzalez-Lopez and Laureano Gonzalez-Vega

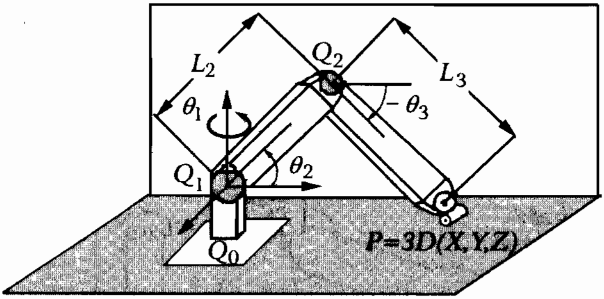
1. Introduction

One of the most basic (although not easy) problems in Robotics is the inverse kinematics problem, i.e., the determination of the values for the internal parameters of a robot manipulator (parameters measured in the motor joints) in order that a particular configuration (position and orientation) of the manipulator tip is reached.

In this project, by studying in detail two concrete examples, we show how to use some of the techniques, introduced in Chapters 2 and 6, to deal with the corresponding algebraic system of equations (real or complex solutions).

2. The ROMIN Manipulator

Let $\mathcal{R}$ be a robot manipulator located in $\mathbb{R}^3$ with three arms and three degrees of freedom:



The three arms are always in the same vertical plane of $\mathbb{R}^3$; θ_1 is the angle measured in the first arm (the base of the robot), which rotates with respect to an axis ($\overrightarrow{Q_0Q_1}$) perpendicular to the ground. Every angle θ_1 determines one vertical plane containing the line Q_0Q_1, where the two last arms of the robot are going to move. The angle θ_2 (respectively, θ_3) is measured from the horizontal plane through Q_1 (respectively, Q_2) to the second arm (respectively, the third arm) in counter-clockwise direction. If $P = (X, Y, Z)$ is a point in $\mathbb{R}^3$ then the following system of equations (once L_2 and L_3 are fixed):

$$\begin{aligned} X &= -\sin(\theta_1)(L_2\cos(\theta_2) + L_3\cos(\theta_3)) \\ Y &= \cos(\theta_1)(L_2\cos(\theta_2) + L_3\cos(\theta_3)) \\ Z &= L_2\sin(\theta_2) + L_3\sin(\theta_3) \end{aligned}$$

has as solutions the values of the internal parameters $(\theta_1, \theta_2, \theta_3)$ such that the extreme point of the robot reaches the point P (remark that the coordinates (X,Y,Z) of P are considered in a coordinate system of $\mathbb{R}^3$ whose origin is in Q_1). The translation of these equations to an algebraic system of equations is made in the usual way:

$$\begin{aligned} X &= -s_1(L_2c_2 + L_3c_3) \qquad & 0 &= s_1^2 + c_1^2 - 1 \\ Y &= c_1(L_2c_2 + L_3c_3) & 0 &= s_2^2 + c_2^2 - 1 \\ Z &= L_2s_2 + L_3s_3 & 0 &= s_3^2 + c_3^2 - 1 \end{aligned} \tag{$\mathcal{T}$}$$

where $s_i = \sin(\theta_i)$ and $c_i = \cos(\theta_i)$.

The optimal solution for this problem, from the symbolic point of view, is to solve the algebraic system of equations without giving concrete values to the parameters L_2 and L_3 or to the coordinates of the point P. For the particular case considered here, a full symbolic solution can be found by hand via the Dynamic Evaluation Method, as described in [4] (see also [5]). The initial system $\mathcal{T}$ is thus decomposed as a union of 3 systems of polynomial inequalities (on the parameters) and equalities:

$$e \overset{\text{def}}{=} X^2 + Y^2 + Z^2 + L_2^2 - L_3^2 \qquad\qquad d^2 \overset{\text{def}}{=} X^2 + Y^2$$

$$\left\{\begin{array}{l} X^2 + Y^2 \neq 0 \\ Z^2 + X^2 + Y^2 \neq 0 \\ ds_1 = -X \\ dc_1 = Y \\ 4L_2^2(Z^2 + d^2)s_2^2 - 4ZL_2es_2 \\ \qquad\qquad +e^2 - 4d^2L_2^2 = 0 \\ 2dL_2c_2 = -2ZL_2s_2 + e \\ L_3s_3 = Z - L_2s_2 \\ L_3c_3 = d - L_2c_2 \end{array}\right. \quad \left\{\begin{array}{l} X = 0 \\ Y = 0 \\ Z \neq 0 \\ c_1^2 + s_1^2 = 1 \\ 2ZL_2s_2 = e \\ c_2^2 + s_2^2 = 1 \\ L_3s_3 = Z - L_2s_2 \\ L_3c_3 = -L_2c_2 \end{array}\right. \quad \left\{\begin{array}{l} X = 0 \\ Y = 0 \\ Z = 0 \\ L_2^2 - L_3^2 = 0 \\ c_1^2 + s_1^2 = 1 \\ c_2^2 + s_2^2 = 1 \\ L_3s_3 = -L_2s_2 \\ L_3c_3 = -L_2c_2 \end{array}\right.$$

Exercise 2.1. Prove that the union of the zeroset of each one of these three systems agrees with the solutions of the system $\mathcal{T}$.

In general, this problem can also be treated by using Gröbner Bases (see [2]), but in this particular example such a method fails when the parameters are specialized to concrete values (in general, the result after specialization need not be a Gröbner Basis). This problem can be avoided by using the "Comprehensive Gröbner Bases" (see [8]), which are more difficult to compute than the usual Gröbner Bases.

Any of these methods, including the Dynamic Evaluation one, produces systems of equations equivalent to the given system in the complex case. After this, the work remains of determining which are the real solutions, or even more interesting, which are the sets of parameters for which there exist real solutions. In the example we are considering, the equations in the three systems are simple enough to allow easily derive the conditions on the parameters assuring that the systems have real solutions: they can be obtained by adding to every system the inequalities:

$$0 \leq s_1^2 \leq 1, \quad 0 \leq s_2^2 \leq 1, \quad 0 \leq s_3^2 \leq 1,$$

which are translated into conditions on X, Y, Z, L_2, and L_3 in every case.

Exercise 2.2. Deduce conditions on X, Y, Z, L_2, and L_3 expressing that each of the three previous systems has real solutions.

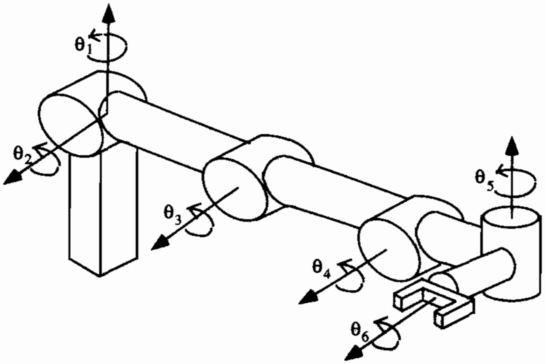
Exercise 2.3. From the union of the three systems, conclude that the set of admissible points, i.e., the points $P = (X, Y, Z)$ reachable for some configuration $(\theta_1, \theta_2, \theta_3)$ of the robot with lenghts L_2 and L_3 is

$$\{(X, Y, Z) \in \mathbb{R}^3 \mid (L_2 - L_3)^2 \leq X^2 + Y^2 + Z^2 \leq (L_2 + L_3)^2\}.$$

Although in the previous particular case a symbolic (and optimal solution) was derived by using Dynamic Evaluation, in general this is not always the case as shown in the next section.

3. The Elbow Manipulator

The elbow manipulator is depicted in the next figure:



Its equations are derived by using the classical matricial form, with the Denavit-Hartenberg parameters specialized for this robot (see [3] and [6]). With this convention we obtain the following twelve equations:

$$\begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_2 & -s_2 & 0 & c_2 \\ s_2 & c_2 & 0 & s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_3 & -s_3 & 0 & c_3 \\ s_3 & c_3 & 0 & s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} c_4 & 0 & -s_4 & c_4 \\ s_4 & 0 & c_4 & s_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} h_{11} & h_{12} & h_{13} & v_1 \\ h_{21} & h_{22} & h_{23} & v_2 \\ h_{31} & h_{32} & h_{33} & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where:

- s_i and c_i represent the sine and cosine of the angle θ_i, $i = 1, \ldots, 6$, and
- the matrix $(h_{i,j}, v_i)_{i,j}$, containing the parameters of this example, characterizes the position and orientation of the manipulator tip.

We add to this initial system the usual equations relating sine and cosine for every angle:

$$\begin{array}{lll} c_6^2 + s_6^2 - 1 = 0 & \quad c_4^2 + s_4^2 - 1 = 0 & \quad c_5^2 + s_5^2 - 1 = 0 \\ c_3^2 + s_3^2 - 1 = 0 & \quad c_2^2 + s_2^2 - 1 = 0 & \quad c_1^2 + s_1^2 - 1 = 0 \end{array}$$

The use of `Recipe VI` (Chapter 2) and `Recipes` CRS$_2$, CRS$_3$ or CRS$_4$, (Chapter 6) allows us, for given values of the parameters $(h_{i,j}, v_i)_{i,j}$, to compute, respectively, the number of different complex solutions and how many of them are real.

Exercise 3.1. Let $\mathcal{O}_i$ $(1 \le i \le 5)$ be the matrices

$$\mathcal{O}_1 = \begin{pmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathcal{O}_2 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathcal{O}_3 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathcal{O}_4 = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} & \frac{2}{3} & \frac{7}{9} \\ \frac{-1}{3} & \frac{-2}{3} & \frac{2}{3} & \frac{2}{9} \\ \frac{2}{3} & \frac{-2}{3} & \frac{-1}{3} & \frac{5}{9} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathcal{O}_5 = \begin{pmatrix} \frac{3}{7} & \frac{6}{7} & \frac{2}{7} & \frac{1}{9} \\ \frac{2}{7} & \frac{-3}{7} & \frac{6}{7} & \frac{2}{9} \\ \frac{6}{7} & \frac{-2}{7} & \frac{-3}{7} & \frac{4}{9} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

representing five different specializations of the parameters for the elbow manipulator (position and orientation of the manipulator tip). If $c(\mathcal{O}_i)$ denotes the number of complex solutions associated to $\mathcal{O}_i$ and $r(\mathcal{O}_i)$ the number of real solutions, use your favourite Computer Algebra System and the `recipes` `VI`, in Chapter 2, and CRS$_2$, CRS$_3$ and CRS$_4$, in Chapter 6, to confirm the results in the following table:

	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	$\mathcal{O}_4$	$\mathcal{O}_5$
$c(\mathcal{O}_i)$	8	8	8	8	8
$r(\mathcal{O}_i)$	0	4	4	8	8

The algorithms to be used in the previous exercise have been very efficiently implemented inside the European projects PoSSo (a Esprit/BRA project) and FRISCO (a Esprit/LTR project). They can be contacted:

- by email at `posso@posso.dm.unipi.it` and
- through the web site `http://extweb.nag.co.uk/projects/FRISCO`,

respectively. Interactive demos and/or software implementations of all these algorithms are available from the following web sites:

- `http://janet.dm.unipi.it/posso_demo.html` for PoSSo, and
- `http://www.loria.fr/~rouillie/docrs/rs/rs.html` for the FRISCO `RealSolving` part.

The example we are considering in this section, the elbow manipulator, is a particular case of the robot class known as 6R-robot manipulator. For this class, by using elimination techniques, a polynomial of degree 16 can be found whose solutions are the values of one of the six internal parameters for the 6R (cf. [7] or [1]). For the particular case of the elbow manipulator, in [6], a closed formula (involving only square roots) can be found giving the six internal parameters in terms of the h_{ij} and v_k; these formulae have been obtained by using ad hoc techniques. For a particular case, this closed form solution does not provide any insight about the number of real solutions before performing the substitution, and the computations showed before help to avoid round-off errors.

References

1. J. Canny and D. Manocha (1994): *Efficient inverse kinematics of general 6R manipulators.* IEEE Transactions on Robotics and Automation 10 (5), 648–657.
2. B. Buchberger (1988): *Applications of Gröbner bases in non-linear computational geometry.* Trends in Computer Algebra. Lecture Notes in Computer Science **296**, 52–80, Springer-Verlag, Berlin Heidelberg New York.
3. J. Denavit and R.S. Hartenberg (1955): *A kinematic notation for lower-pair mechanisms based on matrices.* Journal of Applied Mechanics, 215–221.
4. D. Duval (1995): *Evaluation dynamique et clôture algébrique.* Journal of Pure and Applied Algebra **99**, 267–295.
5. M.-J. Gonzalez-Lopez and T. Recio (1993): *The ROMIN inverse geometric model and the Dynamic Evaluation method.* Computer Algebra in Industry: Problem Solving in Practice, 117–142, John Wiley and Sons, New York.
6. R. P. Paul (1981): *Robot manipulators: Mathematics, Programming and Control.* The MIT Press Series in Artificial Intelligence.

7. M. Raghavan and B. Roth (1989): *Kinematic analysis of the 6R manipulator of general geometry.* Proceedings of the International Symposium on Robotics Research (Tokyo), 314–320.
8. V. Weispfenning (1992): *Comprehensive Gröbner Bases.* Journal of Symbolic Computation **14** (1), 1–30.

Project 4. Quaternion Algebras

Gábor Ivanyos and Lajos Rónyai

1. Introduction

The exercises in this project intend to address the smallest nontrivial case of the problem of finding zero divisors in noncommutative simple algebras. For simplicity we restrict our attention to ground fields F with $\operatorname{char} F \neq 2$. First we outline (in Section 2) how to find a canonical presentation of a four dimensional noncommutative simple algebra over the field F. In Section 3 we establish a link between the problem of computing zero divisors in these algebras and a problem of arithmetical nature: assume that $\alpha, \beta \in F$ are given elements; find a nontrivial solution $(x, y, z) \in F^4$ of the quadratic equation $\alpha x^2 + \beta y^2 - z^2 = 0$. This latter problem is likely to be difficult if F is the field of rational numbers (see [2]). We shall use the notation of Chapter 5. For a detailed exposition of quaternion algebras the reader is referred to [1] and [3].

2. Four Dimensional Simple Algebras

Throughout this section we assume that $\mathcal{A}$ is a noncommutative semisimple algebra over F such that $\dim_F \mathcal{A} = 4$.

Exercise 2.1. Show that $\mathcal{A}$ is simple and $C(\mathcal{A}) = F1_{\mathcal{A}}$. (Hint: Use Wedderburn's structure theorem.)

The next exercise demonstrates that the explicit isomorphism problem is equivalent to finding a *single* zero divisor in $\mathcal{A}$.

Exercise 2.2. Assume that we are given a zero divisor $0 \neq z \in \mathcal{A}$. Suggest an efficient algorithm for finding an explicit isomorphism $\mathcal{A} \cong \mathrm{M}_2(F)$. (Hint: show that $\dim_F \mathcal{A}z = 2$ and consider the action of $\mathcal{A}$ on the left ideal $\mathcal{A}z$.)

Exercise 2.3. Assume that $\mathcal{B} < \mathcal{A}$ is a commutative subfield of $\mathcal{A}$. Show that $\dim_F \mathcal{B} \leq 2$. (Hint: $\mathcal{B}$ acts on $\mathcal{A}$ by multiplication from the left. This makes $\mathcal{A}$ a linear space over $\mathcal{B}$.)

Exercise 2.4. Assume that $\mathcal{B} < \mathcal{A}$ is a commutative subalgebra of $\mathcal{A}$. Prove that $\dim_F \mathcal{B} \leq 2$. (Hint: We may assume that $\mathcal{B}$ is not semisimple, in particular $\mathcal{A}$ contains a zero divisor, therefore $\mathcal{A} \cong \mathrm{M}_2(F)$.)

Exercise 2.5. Prove that for an arbitrary element $u \in \mathcal{A} \setminus F1_{\mathcal{A}}$ the centralizer $C_{\mathcal{A}}(u)$ is the linear subspace of $\mathcal{A}$ generated by $1_{\mathcal{A}}$ and u.

Exercise 2.6. Show that for an arbitrary element $u \in \mathcal{A} \setminus F1_{\mathcal{A}}$ there exists a unique monic polynomial $f(x) = x^2 + \alpha_1 x + \alpha_0 \in F[x]$ of degree 2 such that $f(u) = u^2 + \alpha_1 u + \alpha_0 1_{\mathcal{A}} = 0$.

The polynomial $f(x)$ in Exercise 2.6 is called the *minimal polynomial* of u.

Exercise 2.7. Propose an efficient algorithm for finding an element $u \in \mathcal{A} \setminus F1_{\mathcal{A}}$ such that $u^2 = \alpha 1_{\mathcal{A}}$, where $\alpha \in F$ is a nonzero scalar.

From now on u denotes an element from $\mathcal{A} \setminus F1_{\mathcal{A}}$ such that $u^2 = \alpha 1_{\mathcal{A}}$, where α is a nonzero scalar from F.

Exercise 2.8. Prove that the F-linear map $\phi : \mathcal{A} \to \mathcal{A}$ given as $\phi(v) = uv + vu$ is not bijective. (Hint: Show that $\operatorname{im} \phi \subseteq C_{\mathcal{A}}(u)$.)

Exercise 2.9. Propose an efficient algorithm for finding an element $0 \neq v \in \mathcal{A}$ such that $uv = -vu$. (Hint: Preceding exercise.)

From now on we assume that we are also given an element $v \in \mathcal{A} \setminus 0$ such that $uv = -vu$.

Exercise 2.10. Show that there exists $\beta \in F$ such that $v^2 = \beta 1_{\mathcal{A}}$. (Hint: Use Exercise 2.5 to prove that v^2 is in the subalgebra $\mathcal{B}$ generated by u. Show that the assumption $v^2 \in \mathcal{B} \setminus F1_{\mathcal{A}}$ would contradict $uv = -vu$.)

Assume that $v^2 = \beta 1_{\mathcal{A}}$, where $\beta \neq 0$. (The case $v^2 = 0$ can be treated by exercise 2.2.)

Exercise 2.11. Prove that the elements $1_{\mathcal{A}}, u, v, uv$ form a basis of $\mathcal{A}$. Determine the structure constants of $\mathcal{A}$ with respect to this basis.

3. Quaternion Algebras and Quadratic Forms

Let α, β be nonzero elements of F. Let $H(\alpha, \beta)$ stand for the 4-dimensional F-space generated by $1, u, v, w$. We define (bilinear) multiplication on $H(\alpha, \beta)$ by letting 1 act as identity element and

$$u^2 = \alpha, \quad v^2 = \beta, \quad uv = -vu = w.$$

Exercise 3.1. Show that this operation can be extended to an associative multiplication. (Hint: It suffices to ensure associativity on basis elements.)

We can write the matrices of the (left) regular representation of $H(\alpha,\beta)$ in terms of the basis $1, u, v, w$ as

$$u \mapsto \begin{pmatrix} 0 & 1 & & \\ \alpha & 0 & & \\ & & 0 & 1 \\ & & \alpha & 0 \end{pmatrix}, v \mapsto \begin{pmatrix} & & 1 & 0 \\ & & 0 & -1 \\ \beta & 0 & & \\ 0 & -\beta & & \end{pmatrix},$$

$$w \mapsto \begin{pmatrix} & & 0 & 1 \\ & & -\alpha & 0 \\ 0 & \beta & & \\ -\alpha\beta & 0 & & \end{pmatrix}.$$

In the preceding section we gave an effective proof of the fact that every four dimensional noncommutative simple algebra over F is of the form $H(\alpha,\beta)$ for suitable scalars $\alpha, \beta \in F$. The next exercise demonstrates that the converse also holds.

Exercise 3.2. Show that $H(\alpha,\beta)$ is simple and $C(H(\alpha,\beta)) = F1$. (Hint: Using the trace form of the regular representation prove that $H(\alpha,\beta)$ is semisimple.)

For an element $x = \gamma_1 1 + \gamma_u u + \gamma_v v + \gamma_w w$ define the conjugate x^* of x as $x^* = \gamma_1 1 - \gamma_u u - \gamma_v v - \gamma_w w$.

Exercise 3.3. Let $x, y \in H(\alpha,\beta)$ and $\gamma, \delta \in F$. Show that

(i) $(\gamma x + \delta y)^* = \gamma x^* + \delta y^*$;
(ii) $(xy)^* = y^* x^*$;
(iii) $(x^*)^* = x$;
(iv) $x^* = x \Leftrightarrow x \in F1$;
(v) $xx^* = x^* x \in F1$;
(vi) x is a zero divisor $\Leftrightarrow xx^* = 0$.

Exercise 3.4. Show that finding a zero divisor in $H(\alpha,\beta)$ is equivalent to finding a nontrivial solution (z_1, z_u, z_v, z_w) in F^4 to the equation

$$z_1^2 - \alpha z_u^2 - \beta z_v^2 + \alpha\beta z_w^2 = 0.$$

Exercise 3.5. Show that finding a zero divisor in $H(\alpha,\beta)$ is equivalent to finding a nontrivial solution (x, y, z) in F^3 to the equation

$$\alpha x^2 + \beta y^2 - z^2 = 0.$$

(Hint: Show that $\gamma_1 1 + \gamma_u u + \gamma_v v + \gamma_w w \in H(\alpha,\beta)$ is nilpotent iff $\gamma_1 = 0$ and $\alpha\gamma_u^2 + \beta\gamma_v^2 - \alpha\beta\gamma_w^2 = 0$. Set $x = \gamma_v/\alpha$, $y = \gamma_u/\beta$, and $z = \gamma_w$.)

References

1. R. S. Pierce (1982): *Associative Algebras*, Springer-Verlag, Berlin.
2. L. Rónyai (1988): *Zero Divisors in Quaternion Algebras*, Journal of Algorithms **9**, 494–506.
3. W. Scharlau (1985): *Quadratic and Hermitian Forms*, Springer-Verlag, Berlin.

Project 5. Explorations with the Icosahedral Group

Arjeh M. Cohen, Hans Cuypers, and Remko Riebeek

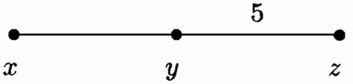
1. Introduction

In Project 6 we have encountered a way to construct groups via a permutation representation. In the early seventies this has been one of the main tools in constructing sporadic simple groups. However, the permutation representations of the large sporadic simple groups like the so-called Monster and Baby-Monster have too high degree to put them on a computer, see the Atlas [1]. For these groups one has to use different methods. Many of these large sporadic simple groups, including the Monster (see [4]), have been constructed as a matrix group. In this project we will show by means of a small example how one may proceed to construct a group as a matrix group.

The example we will work with is the *Coxeter group* $W(H_3)$. This is the group given by the following presentation:

$$\langle x, y, z \mid x^2, y^2, z^2, (xy)^3, (yz)^5, (xz)^2 \rangle.$$

Usually these relations are summarized by the Coxeter diagram of type H_3:



A priori, the group $W(H_3)$ may be infinite, finite, or even trivial. In this project we will determine the precise structure of $W(H_3)$. In Section 2 we construct a three-dimensional real representation of $W(H_3)$ from which we deduce that the group $W(H_3)$ has a quotient that is the automorphism group of the *icosahedron*. Then, in Section 3, we perform coset enumeration to determine the order and exact structure of the group $W(H_3)$. In particular, we will show that $W(H_3)$ *is* isomorphic to the automorphism group of the icosahedron; for this reason the group is also referred to as the *icosahedral group*. Finally, the permutation representation obtained from the coset enumeration will be exploited to find several other linear representations of $W(H_3)$.

Although the group $W(H_3)$ falls within the realm of Coxeter groups (see for example [2]), we will not make use of the vast machinery developed for such groups in this project. For representation theory in general, see [3].

2. Three-Dimensional Representations for $W(H_3)$

Consider $W = \langle x, y, z \mid x^2, y^2, z^2, (xy)^3, (yz)^5, (xz)^2 \rangle$, the Coxeter group $W(H_3)$ of type H_3. We investigate the three-dimensional real linear representations of this group, i.e., the homomorphisms of W into the group $GL_3(\mathbb{R})$.

Exercise 2.1. Show that x and z commute, and that the elements x, y, z are conjugate in W.

Suppose $\phi : W \to GL_3(\mathbb{R})$ is a homomorphism. By V we denote the vector space $\mathbb{R}^3$. We assume that W acts from the right on the vectors of V.

Exercise 2.2. Suppose $\phi(x) = \phi(z)$. Prove that $\phi(x) = \phi(y) = \phi(z)$ and that there is a one-dimensional subspace of V invariant under $\phi(W)$. In particular, the elements $\phi(x)$, $\phi(y)$ and $\phi(z)$ either all induce scalar multiplication with -1 or are all trivial on this subspace.

From now on we assume that x and z are mapped to distinct elements in $GL_3(\mathbb{R})$. In particular, the elements x, y, z are mapped to non-identity matrices!

Exercise 2.3. Use $\phi(x)^2 = id$ to show that $V = \ker(\phi(x) - id) \oplus \ker(\phi(x) + id)$. Then show that, without loss of generality, we may assume that we are in one of the following cases:

$$\text{(I)}\ \phi(x) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \phi(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \text{ or}$$

$$\text{(II)}\ \phi(x) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \phi(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Exercise 2.4. Show that for every representation ϕ, there is also a representation ψ of W with $\psi(u) = -\phi(u)$ for $u \in \{x, y, z\}$.

The above exercise indicates that to find all representations we only have to consider representations ϕ as in I and multiply them with -1 to obtain those of case II. Let us do so.

It remains to find the matrix $(y_{i,j})_{1 \le i,j \le 3}$ representing $\phi(y)$. The matrices for $\phi(x)$ and $\phi(z)$ as in case I above can be chosen by fixing a basis of common eigenvectors of $\phi(x)$ and $\phi(z)$. But there is still some freedom in this choice: without loss of generality we can replace the matrix $(y_{i,j})_{1 \le i,j \le 3}$ for $\phi(y)$ with a conjugate of $(y_{i,j})_{1 \le i,j \le 3}$ by an invertible diagonal matrix

$$\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \rho \end{pmatrix}.$$

Exercise 2.5. Prove that (after conjugation) we can assume that $y_{i,j}$ is either 0 or equal to $\pm y_{j,i}$. If we allow complex λ, μ, and ρ, then even $y_{i,j} = y_{j,i}$. Moreover, we can assume that $y_{i,j} \geq 0$ for $i < j$.

Exercise 2.6. The trace of $\phi(y)$ equals 1. Why? This gives us the following equation:

$$y_{1,1} + y_{2,2} + y_{3,3} = 1.$$

What is the order of $\phi(xy)$? What are the complex eigenvalues of $\phi(xy)$? Prove that the trace of $\phi(xy)$ equals 0 and give the resulting equation.

Combine these equations to obtain that $y_{1,1} = -1/2$.

Exercise 2.7. Set up equations for the entries $y_{i,j}$ corresponding to the other information available for $\phi(y)$:

1. $\phi(y)^2 = 1$,
2. $\phi(x)\phi(y)\phi(x) = \phi(y)\phi(x)\phi(y)$,
3. $\phi(y)\phi(z)\phi(y)\phi(z)\phi(y) = \phi(z)\phi(y)\phi(z)\phi(y)\phi(z)$.

Perform a Gröbner basis computation and solve the system of equations (and inequalities) for the parameters $y_{i,j}$ obtained so far.

Check that this computation leads to 2 possible solutions for the matrix $\phi(y)$.

Exercise 2.8. Prove that the two representations we have found in the previous exercises are really distinct, i.e., there is no element $g \in GL(\mathbb{R}^3)$ conjugating one representation to the other. (Hint: consider the trace of $\phi(yz)$.)

Exercise 2.9. What group do $\phi(x)$ and $\phi(y)$ generate? What is its order? Same questions for the group generated by $\phi(y)$ together with $\phi(z)$.

Fix a linear representation ϕ of type I as found above. The elements $\phi(x)$, $\phi(y)$ and $\phi(z)$ are reflections on $\mathbb{R}^3$. Let v be a nontrivial vector in $\mathbb{R}^3$ in the intersection of the two reflection hyperplanes for $\phi(y)$ and $\phi(z)$. Consider the orbit $\mathcal{I}$ of v under the group $\phi(W)$.

Exercise 2.10. Show that $\mathcal{I}$ contains 12 vectors forming the vertices of an icosahedron.

Check that $\phi(W)$ is contained in the automorphism group of this icosahedron.

Prove that $\phi(W)$ is actually the full automorphism group of this icosahedron. (Here you might use the same methods as in Sections 2 and 3 of Project 6.)

Exercise 2.11. A linear representation of a group G into $\mathrm{GL}(V)$ is called *irreducible* if V contains no nontrivial proper G-invariant subspace.

Show that ϕ is irreducible.

3. Coset Enumeration

In this section we concentrate on the order of the group $W = W(H_3)$. For this purpose we use the Todd-Coxeter coset enumeration algorithm (see Chapter 8). Therefore we have to fix a subgroup H of W with respect to which we do the coset enumeration. There are two obvious candidates for H: the group $D = \langle x, y\rangle$, which is a quotient of the dihedral group of order 6 and the group $I = \langle y, z\rangle$, which is a quotient group of the dihedral group of order 10.

Exercise 3.1. Use the results of the above section to conclude that the group D is isomorphic to a dihedral group of order 6, and that I is isomorphic to a dihedral group of order 10.

We use GAP [5] to do the coset enumerations with respect to the subgroups I and D of W and to construct the permutation representations of W on the resulting cosets.

```
x:=AbstractGenerator("x");
y:=AbstractGenerator("y");
z:=AbstractGenerator("z");
W:=Group(x,y,z);
W.relators:=[x^2,y^2,z^2,(x*z)^2,(x*y)^3,(y*z)^5];
I:=Subgroup(W,[y,z]);
Iperm:=OperationCosetsFpGroup(W,I);
# Permutation action on the cosets of I
D:=Subgroup(W,[x,y]);
Dperm:=OperationCosetsFpGroup(W,D);
# Permutation action on the cosets of D
```

Exercise 3.2. Perform the coset enumeration and deduce from this the order of W.

Exercise 3.3. Show that both the cyclic group $\mathbb{Z}_2$ of order 2 and the alternating group A_5 are quotients of W, and conclude that $W \simeq \mathbb{Z}_2 \times A_5$.

4. The Permutation Representation of W on the Cosets of I

Let W be as in the previous section and consider the permutation action of W on the 12 cosets of $I = \langle y, z\rangle$. As we have seen in 2.10, these 12 cosets can be viewed as the vertices of an icosahedron. We will construct this icosahedron.

Let P be the set of 12 cosets of I in W. This set P will be the point set of a graph isomorphic to the icosahedron. The group W acts naturally on the set $P \times P$, by $(\alpha, \beta)w = (\alpha w, \beta w)$ for all $w \in W$. The *orbitals* of W

acting on P are the orbits of W acting on $P \times P$ in this way. The directed graph associated to an orbital has as vertices the points of P and as edges the ordered pairs in the orbital. If, for each edge (α, β) of this graph, also the reversed pair (β, α) is an edge, we consider the undirected graph with edges $\{\alpha, \beta\}$ where (α, β) is in the orbital.

Exercise 4.1. Show that W has four orbitals on the cosets of I, of size 12, 12, 60, and 60, respectively.

Consider the graph belonging to one of the orbitals of size 60. Draw the (undirected) graph and verify that it is isomorphic to the graph of the icosahedron.

In GAP we can use:

```
pairs:=Tuples([1..12], 2);
Orbits(Iperm, pairs, OnPairs);

#load the package GRAPE for working with graphs
RequirePackage("grape");

#We construct the graph with edge set the orbital of (1,2).

gamma:=EdgeOrbitsGraph(Iperm, [1,2], 12);
```

We can turn the permutation representation of W on the 12 (right) cosets of I into a linear representation: identifying the 12 cosets (denoted by 1 up to 12) with 12 basis elements of a 12-dimensional (real) vector space, say $V = \langle e_1, \ldots, e_{12} \rangle$, the action of an element of W on V is given by linear extension of the permutation action on the 12 basis vectors. This yields a 12-dimensional linear real representation of W. In GAP this representation is constructed as follows (at the moment we just define V over the rationals):

```
VV:=[];
for i in [1..12] do VV[i]:=[]; od;
for i in [1..12] do for j in [1..12] do VV[i][j]:=0; od; od;
for i in [1..12] do VV[i][i]:=1; od;
V:=RowSpace(Rationals,VV);

# Now we define the matrices of x, y and z on V.
R1mat:=[];R2mat:=[];R3mat:=[];
for i in [1..12] do
  R1mat[i]:=[]; R2mat[i]:=[]; R3mat[i]:=[];
od;
for i in [1..12] do
  for j in [1..12] do
    R1mat[i][j]:=0; R2mat[i][j]:=0; R3mat[i][j]:=0;
    od;
  od;
  for i in [1..12] do
   R1mat[i][i^Iperm.generators[1]]:=1;
   R2mat[i][i^Iperm.generators[2]]:=1;
   R3mat[i][i^Iperm.generators[3]]:=1;
  od;
  Wmat:=Group(R1mat,R2mat,R3mat);
```

In the sequel we will show that the 3-dimensional real representation of Section 2, in which there is a natural representation of the icosahedron, is a quotient of V. To find this quotient we first have to find W invariant subspaces of V.

In the natural 3-dimensional representation of W, each vertex v of the icosahedron has an opposite vertex $-v$. This observation leads us to the following invariant subspace: The twelve points in P can be divided into six pairs $\{i, j(i)\}$, such that i and $j(i)$ are at maximal distance in the graph of the icosahedron. Let $U = \langle e_i + e_{j(i)} \mid i = 1, \dots, 6\rangle$. The construction of U can be carried out in GAP as follows:

```
  Blocks(Iperm,[1..12]);

  # The pairs of opposite vertices in the icosahedron
  # are of the form {i,13-i} in our example
  # and for U we find:
  U:=Subspace(V,[[1,0,0,0,0,0,0,0,0,0,0,1],
                 [0,1,0,0,0,0,0,0,0,0,1,0],
                 [0,0,1,0,0,0,0,0,0,1,0,0],
                 [0,0,0,1,0,0,0,0,1,0,0,0],
                 [0,0,0,0,1,0,0,1,0,0,0,0],
                 [0,0,0,0,0,1,1,0,0,0,0,0]]);
  K:=V/U;
```

Exercise 4.2. Check that U is a W-invariant subspace of V of dimension 6.

Find a one-dimensional subspace, say J, in U that is also W-invariant. It is possible to show that the quotient space U/J is an irreducible representation of W of degree (= dimension of the space) five.

Since U is a W-invariant subspace of V, the group W also acts as a linear group on the 6-dimensional quotient space V/U.

To obtain the natural module for the icosahedral group we have to find a second W-invariant subspace of V containing U and of dimension 9. To do so we use another relation between the points of the icosahedron.

Exercise 4.3. Let v be a vertex of the icosahedron in its natural 3-dimensional representation. Let v_i, $i = 1, \dots, 5$ be the five neighbouring vertices of v. Then there is an $\alpha \in \mathbb{R}$ such that

$$\alpha v = \sum_{i=1}^{5} v_i.$$

Determine α.

Suppose for each $i \in \{1, \dots, 12\}$ the 5 neighbours of j are equal to $j(i)$, $i = 1, \dots 5$. Let $X = \langle \alpha e_j - \sum_{i=1}^{5} e_{j(i)} \mid j \in \{1, \dots, 12\} \rangle$. Then X is a W-invariant subspace of V. What is its dimension?

Use X to construct a 3-dimensional quotient of V in which the icosahedron $\mathcal{I}$ is naturally embedded.

All this means that till now we have constructed irreducible representations of degrees 1 (see 2.2) 3, 3, and 5. Since our group W is isomorphic to $\mathbb{Z}_2 \times \mathrm{A}_5$, we also have constructed such representations for the subgroup A_5 of W. Actually, we have not explicitly shown that these representations are irreducible, but that can be easily checked, even if we restrict them to the subgroup A_5 of W.

We call two linear representations of a group G *equivalent* if one can be obtained from the other by conjugation with an invertible linear transformation. Representation theory, see [3], tells us that the group G has as many inequivalent complex linear representations as the number of its conjugacy classes of elements. Moreover, if these inequivalent irreducible representations are of degree d_i, where $i = 1, \dots, k$ (the number of conjugacy classes of G), then

$$\sum_{i=1}^{k} d_i^2 = |G|.$$

Now A_5 has order 60 and contains five distinct conjugacy classes. So, since $60 - 1 - 9 - 9 - 25 = 16$, we only need to construct an irreducible four-dimensional representation of A_5 and, up to equivalence, our list of irreducible linear representations for A_5 is complete.

Exercise 4.4. The group A_5 acts naturally as a permutation group on 5 elements. Construct a 5-dimensional real vector space Y and a linear representation of A_5 on Y using this permutation action.

Find a 4-dimensional quotient space of Y on which A_5 acts irreducibly.

Exercise 4.5. Determine, up to equivalence, all irreducible representations of the group W.

Exercise 4.6. Construct the natural representation in 3-dimensional real space of the dodecahedron, starting with the permutation action of W on the 20 right cosets of D.

References

1. J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson (1985): *ATLAS of Finite Groups*, Clarendon Press, Oxford.
2. J. E. Humphreys (1990): *Reflection Groups and Coxeter Groups*, Cambridge Studies in Advanced Mathematics **29**, Cambridge University Press.
3. W. Fulton and J. Harris (1991): *Representation Theory: a First Course*, Graduate Texts in Mathematics **129**, Springer-Verlag, New York Berlin Heidelberg.
4. R. L. Griess: The Friendly Giant, *Inventiones Math.* **69** (1982), 1–102.
5. M. Schönert et al. (1994): GAP – *Groups, Algorithms and Programming*, version 3, release 4, Lehrstuhl D für Mathematik, RWTH Aachen.

Project 6. The Small Mathieu Groups

Hans Cuypers, Leonard H. Soicher, and Hans Sterk

1. Introduction

In this project we use the tools and techniques from Chapter 8 to construct the small Mathieu groups M_{10}, M_{11} and M_{12}. These groups were discovered by the French mathematician Émile Mathieu (1835–1890), who also discovered the large Mathieu groups M_{22}, M_{23} and M_{24}. See [9, 10, 11]. They are remarkable groups: for example, apart from the symmetric and alternating groups, M_{12} and M_{24} are the only 5-transitive permutation groups. The group M_{10} has a normal subgroup of index 2 isomorphic to A_6. The other five groups are among the 26 sporadic simple groups, occurring in the classification of finite simple groups. After Mathieu's discovery of these five sporadic simple groups it took almost a century before the sixth sporadic simple group was found.

In fact, many of the algorithms discussed in Chapter 8, and more sophisticated versions of these algorithms, were developed by C. Sims and others in the late sixties and early seventies as part of the construction and study of various sporadic simple groups. For example, one of these 26 sporadic groups, the so-called O'Nan group, was constructed by Sims as a permutation group on 122760 points by giving a generating set of permutations in S_{122760}. Algorithms like those of Chapter 8 helped Sims to identify the group generated by these permutations as the O'Nan group. A more recent construction of the O'Nan group, using coset enumeration, is given in [17].

In this project we want to give the reader some of the flavour of how one may construct a group. For this purpose we consider the small Mathieu groups. There are several ways to describe the Mathieu groups. They appear for instance as automorphism groups of the Golay codes, see Project 7. In this chapter we will find generating permutations for the small Mathieu groups with the help of design theory (following Lüneburg [8]). Moreover we investigate the 2-transitive subgroups of M_{12} and some related graphs and designs. Finally, we classify some graphs, and in the process find a presentation by generators and relators for the automorphism group $M_{12}{:}2$ of M_{12}. This graph classification and presentation are previously unpublished results of the second author. In this project we use the algorithms discussed in 8. For the convenience of the reader we have added GAP-code (see [15]) to perform these computations.

We use ATLAS notation [2] for group structures. Thus $L_2(11)$ is the simple group $PSL_2(11)$, n denotes the cyclic group of that order, and, if p

is prime, p^n denotes the elementary abelian group of that order. A group (of shape) $A.B$ is an arbitrary extension of A by B (A is a normal subgroup of $A.B$ and the quotient by A is isomorphic to B), $A{:}B$ is a split extension, and $A{\cdot}B$ is a non-split extension.

We remark that much useful information on the Mathieu groups is contained in [4, Chapter 11]. Further information on many of the permutation representations described in this chapter can be found in [14].

2. The Affine Plane of Order 3

A *design* $\Delta = (P, B)$ is a set P of *points* together with a collection B of subsets of P, called *blocks*. A design is called a t-(v, k, λ) design if it contains exactly v points, if each block contains exactly k points, and if any set of t points is contained in exactly λ blocks.

The *automorphism group* $\mathrm{Aut}(\Delta)$ of a design Δ is the subgroup of the symmetric group $\mathrm{Sym}(P)$ of the point set consisting of those permutations that map blocks to blocks.

2.1. The Affine Plane of Order 3. In this subsection we discuss 2-(9,3,1) designs. An example of such a design is the following: as points we take the vectors of the vector space $GF(3)^2$, where $GF(3)$ denotes the field with 3 elements. The blocks (sometimes also called *lines*) are the triples of points contained in a coset of a 1-dimensional subspace. Indeed, there are 9 points in the design; any block consists of 3 points and any pair of points is in a unique coset of a 1-dimensional subspace.

It is not hard to show that this is, up to isomorphism, the only 2-(9, 3, 1) design. For this reason, the design is also called the *affine plane of order* 3.

Let us denote this unique design by Θ. It is displayed in Figure 2.2. Its points and lines are encoded as follows:

points	
(0,0)	1
(1,0)	2
(−1,0)	3
(0,1)	4
(1,1)	5
(−1,1)	6
(0,−1)	7
(1,−1)	8
(−1,−1)	9

lines					
1	2	3	1	6	8
1	5	9	1	4	7
2	6	7	2	4	9
2	5	8	3	4	8
3	5	7	3	6	9
4	5	6	7	8	9

2.2. Of course, the automorphism group of the design contains the group of translations, a group of order 9. The stabilizer of $(0, 0)$ contains the group $GL_2(3)$. In particular, the automorphism group of the design contains a group isomorphic to the split extension $3^2{:}GL_2(3)$. The order of this group is $3^2 \cdot 48 = 432$.

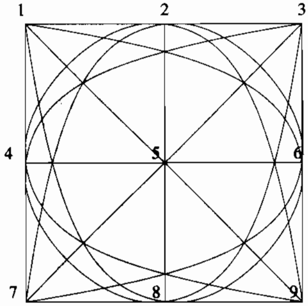


Fig. 2.1. The affine plane of order 3.

As indicated above, we will consider the automorphism group of Θ as a subgroup of the permutation group on the set P of points. With the labeling given in Figure 2.2 this means it is a subgroup of S_9.

An easy check shows that the following permutations are contained in $H := \mathrm{Aut}(\Theta)$:

$$\begin{aligned}
a &= (1,2,3)(4,5,6)(7,8,9), && \text{a translation}\\
b &= (1,4,7)(2,5,8)(3,6,9), && \text{a translation}\\
c &= (2,9,3,5)(4,6,7,8), &&\\
d &= (2,7,3,4)(5,8,9,6), &&\\
e &= (5,7)(4,9)(6,8), &&\\
f &= (4,7)(5,8)(6,9). &&
\end{aligned}$$

Exercise 2.3. 1. Prove that $[1,2,4]$ is a base for the full automorphism group of Θ, and use this to show that this automorphism group has order $9 \cdot 8 \cdot 6 = 432$.

2. Show that the set $\{a,b,c,d,e,f\}$ is a strong generating set for H with respect to the base $[1,2,4]$.
3. Prove that H is the full automorphism group of the affine plane Θ.
4. Show that H is 2-transitive on the points of Θ, but not 3-transitive.

The subgroup $G := \langle a,b,c,d \rangle$ of H is also 2-transitive on the points of Θ. It is a normal subgroup of H of order 72. This is easily checked in GAP:

```
a:=(1,2,3)(4,5,6)(7,8,9); b:=(1,4,7)(2,5,8)(3,6,9);
c:=(2,9,3,5)(4,6,7,8);    d:=(2,7,3,4)(5,8,9,6);
e:=(5,7)(4,9)(6,8);       f:=(4,7)(5,8)(6,9);

H:=Group(a,b,c,d,e,f);
Print("H has size ", Size(H), "\n");
G:=Subgroup(H,[a,b,c,d]);
Orbs:=Orbits(G,[1..9]);
if Size(Orbs)=1
  then  Print("G is transitive", "\n");
        G1:=Stabilizer(G,1);
        Orbs1:=Orbits(G1,[2..9]);
        if Size(Orbs1)=1
          then  Print("G is 2-transitive","\n");
          else  Print("G is not 2-transitive","\n");
        fi;
  else  Print("G is not transitive","\n");
fi;

Print("G has size ",Size(G),"\n");
Print("G is normal in H is ", IsNormal(H,G),"\n");
```

3. A 3-$(10,4,1)$ Design and the Mathieu Group M_{10}

In this section we will construct a 3-$(10,4,1)$ design – together with an automorphism group acting 3-transitively on the 10 points – as an extension of the 2-$(9,3,1)$ design Θ.

3.1. Θ as a Residue. Suppose $\Delta = (P, B)$ is a 3-$(10,4,1)$ design. Then the number of blocks in B equals $10 \cdot 9 \cdot 8/(4 \cdot 3 \cdot 2) = 30$. Moreover, each point is on 12 blocks. Fix some point p of Δ, and consider the *residue* Δ_p of Δ at the point p, where

$$\Delta_p = (P \setminus \{p\}, \quad B_p = \{b \setminus \{p\} \mid \ b \in B \text{ and contains } p\}).$$

Then Δ_p is a 2-$(9,3,1)$ design, and hence isomorphic to the affine plane Θ discussed above. To be specific, take $p = 10$ and identify Δ_{10} with this affine plane. The 12 blocks of Δ on $p = 10$ are then the sets $\{10\} \cup b$ where b is a block of Θ.

Next we want to show how to reconstruct Δ from the design Θ. For that purpose we still have to determine the remaining $30 - 12 = 18$ blocks.

3.2. 4-Arcs. A block of Δ not containing 10 consists of 4 points of Θ meeting any block of Θ in at most 2 points. Any set of 4 points of Θ with this property is called a 4-*arc*. The number of 4-arcs in Θ is equal to $54 = 9 \cdot 8 \cdot 6 \cdot 3/(4 \cdot 3 \cdot 2 \cdot 1)$.

By using the orbit-algorithm we can easily check that H is transitive on the set of 4-arcs of Θ. However, under the action of the smaller group G this orbit splits into 3 orbits of size 18.

```
Arcs:=Orbit(H,[1,2,4,5],OnSets);
if Length(Arcs)=54
  then  Print("The arcs are in one H-orbit of length 54\n");
fi;

G_Arcs_orbits:=Orbits(G,Arcs,OnSets);
Print("This orbit splits in ", Length(G_Arcs_orbits),
 " orbits of length " );
for orbit in G_Arcs_orbits
```

```
    do  Print(Length(orbit), " resp. ");
    od;
  Print("\n");
```

3.3. A 3-Design and Its Automorphism Group. Let Δ be the following design: the point set is $\{1,\dots,10\}$; the blocks of Δ are the 12 sets $\{10\}\cup b$, where b is a block of Θ, and the eighteen 4-arcs in the G-orbit of $\{1,2,4,5\}$. We check that Δ is a 3-(10, 4, 1) design.

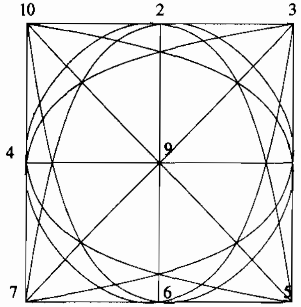


Fig. 3.1. The affine plane Δ_1.

Fix the point 1 and consider the residue Δ_1. The 9 points and 12 blocks form a 2-(9, 3, 1) design isomorphic to Θ, see Figure 3.3. The automorphism group of Δ_1 contains the translation

$$g = (10,2,3)(4,9,8)(7,6,5).$$

It is easily checked, for example with GAP, that the block set of Δ is invariant under g, so $g \in \mathrm{Aut}(\Delta)$.

Let M_{10} be the subgroup of S_{10} generated by G and the element g. Then M_{10} is transitive on the 10 points of Δ. Hence at each point p of Δ the residual design is an affine plane. But then Δ itself is indeed a 3-(10, 4, 1) design.

An order computation yields that $|M_{10}| = 10\cdot 9\cdot 8 = 720$:

```
g:=(10,2,3)(4,9,8)(7,6,5);
m10:=Group(a,b,c,d,g);
Print("m10 has size ",Size(m10),"\n");
```

Thus the point stabilizer of 10 in M_{10} has order 72. Since $|G| = 72$ and since G is contained in the point stabilizer of 10, we conclude that these two groups coincide. In particular, as G is 2-transitive on $\{1,\dots,9\}$, the group M_{10} is 3-transitive on $\{1,\dots,10\}$; in particular, it is transitive on the 30 blocks. It is called the *Mathieu group* of degree 10.

Exercise 3.4. 1. Find a base and strong generating set for M_{10}.

2. Show that M_{10} has index 2 in the full automorphism group of Δ.

The design Δ is, up to isomorphism, the unique 3-$(10, 4, 1)$ design, as can be proven easily. See [6].

4. The Groups M_{11} and M_{12}

The preceding procedure gives us three ways, corresponding to the three choices for the G-orbit on the 4-arcs of Θ, to complete the design Θ to a 3-$(10, 4, 1)$ design Δ. As stated above all three ways lead to isomorphic designs; here it follows directly from the fact that H normalizes G and permutes the 3 choices of 18 blocks. However, it also shows how we may proceed to extend Δ to a 4-$(11, 5, 1)$ design and even to a 5-$(12, 6, 1)$ design. We will construct a 4-$(11, 5, 1)$ design and a 5-$(12, 6, 1)$ design with the help of a 4- and a 5-transitive group.

4.1. The Groups. Let $O_1 = \{1, 2, 4, 5\}^G$, $O_2 = \{1, 2, 4, 8\}^G$ and $O_3 = \{1, 2, 4, 6\}^G$ be the three orbits of G on the 4-arcs of Θ. For $i = 1, 2, 3$, extend Θ to a design Δ^i with point set $\{1, \ldots, 9\} \cup \{9 + i\}$, and with blocks the sets $b \cup \{9 + i\}$ (where b runs through the blocks of Θ) and the sets in O_i. Then both Δ^2 and Δ^3 are 3-$(10, 4, 1)$ designs just as $\Delta^1 = \Delta$.

As before we can prove that $g_2 = (11, 2, 3)(4, 6, 9)(7, 5, 8)$ and $g_3 = (12, 2, 3)(4, 8, 5)(7, 9, 6)$ are automorphisms of Δ^2 and Δ^3, respectively.

Exercise 4.2. Draw the two affine planes Δ^2_1 and Δ^3_1 and check that g_2 and g_3 induce translations on these planes.

Consider the groups $M_{11} = \langle M_{10}, g_2 \rangle$ and $M_{12} = \langle M_{11}, g_3 \rangle$. Since g_2 moves 11 and g_3 moves 12, these groups are transitive on $\{1, \ldots, 11\}$ and $\{1, \ldots, 12\}$, respectively.

Order calculations reveal that

$$|M_{11}| = 11 \cdot |M_{10}| = 11 \cdot 10 \cdot 9 \cdot 8 = 7920,$$

and

$$|M_{12}| = 12 \cdot |M_{11}| = 12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 = 95040.$$

Thus, as M_{10} is 3-transitive, M_{11} is 4-transitive and M_{12} 5-transitive. The groups M_{11} and M_{12} are called the *Mathieu groups* of degrees 11 and 12, respectively.

For later use we define these groups in GAP:

```
g2:=(11,2,3)(4,6,9)(7,5,8); g3:=(12,2,3)(4,8,5)(7,9,6);
m11:=Group(a,b,c,d,g,g2); m12:=Group(a,b,c,d,g,g2,g3);
```

4.3. The Designs. With the groups M_{11} and M_{12} at hand we construct a 4-$(11, 5, 1)$ and a 5-$(12, 6, 1)$ design as follows. As point set we take the sets $\{1, \dots, 11\}$ and $\{1, \dots, 12\}$, respectively. The set of blocks of the 11-design is the M_{11}-orbit of $\{10, 11, 1, 2, 3\}$. For the 12-design we take as block set the M_{12}-orbit of $\{10, 11, 12, 1, 2, 3\}$.

Exercise 4.4. 1. Prove that the designs constructed above are indeed 4-$(11, 5, 1)$ and 5-$(12, 6, 1)$ designs with automorphism groups M_{11}, M_{12}, respectively.

2. Prove that there does not exist a 6-(13,7,1) design by counting the number of blocks.
3. Use this to show that there is no 6-transitive group on 13 elements with point stabilizer isomorphic to M_{12}.

Remark 4.5. The above construction of the small Mathieu groups also reveals a uniqueness proof of the $2 + i$-$(9 + i, 3 + i, 1)$ designs admitting a $(2 + i)$-transitive automorphism group.

5. Two 2-Transitive Subgroups of M_{12}

In this section we are interested in highly transitive subgroups of the automorphism groups of the 5-(12,6,1) design Γ. In particular, we determine the 2-transitive subgroups G of M_{12} acting on the 12 points of Γ.

Exercise 5.1. Let G be a subgroup of S_{12}. Prove that G acts 2-transitively on $\{1, \dots, 12\}$ if and only if it contains (at least) two elements of order 11 with distinct support.

The above exercise will be our starting point for the classification of all 2-transitive subgroups of M_{12}. We first fix the element

$$s = (2, 8, 12, 6, 4, 7, 11, 10, 9, 3, 5)$$

of order 11 in M_{12}, and denote by S the Sylow subgroup of order 11 in M_{12} generated by s. By N we denote the normalizer of S in M_{12}.

Exercise 5.2. Check that the element s is indeed an element of M_{12}. Determine 1) generators for N, 2) the order of N and 3) its structure.

```
s:=(2,8,12,6,4,7,11,10,9,3,5);
Print("s is an element of m12 ? ", s in m12,"\n");
S:=Subgroup(m12,[s]);
N:=Normalizer(m12,S);
Print(N);
Print(" N has size ", Size(N), "\n");
```

Exercise 5.3. Use Sylow's Theorem to show that any 2-transitive subgroup of M_{12} contains a conjugate of $\langle S, S^g \rangle$, where g runs through a set of representatives for double cosets in $\{NgN \mid g \in M_{12}\}$.

In the following GAP-code we determine the proper 2-transitive subgroups G of M_{12} of the form $\langle s, s^g \rangle$ (where s^g denotes $g^{-1}sg$), where g runs through a set of representatives for the double cosets in $\{NgN \mid g \in M_{12}\}$:

```
Twotrans:=[];  # list of proper 2-transitive subgroups found so far
D:=DoubleCosets(m12,N,N);
for i in [1..Length(D)]
    do  g:=Representative(D[i]);
        G:=Subgroup(m12,[s,s^g]);
        if 1^g <> 1 and Size(G) <> 95040 and not (G in Twotrans)
          then  Add(Twotrans,G);
                Print(G,"\n", " G has size ", Size(G),"\n");
        fi;
    od;
```

Exercise 5.4. Run the above program to show that there are only two proper 2-transitive subgroups G of M_{12} of the form $\langle s, s^g \rangle$. One has order 7920, the other order 660.

5.5. 2-Transitive Subgroups of Order 7920. Suppose G is a 2-transitive proper subgroup of M_{12} of order 7920 obtained by running the above program. Then G has the same order as M_{11}. This is not a coincidence. We will see that the groups G and M_{11} are isomorphic. To this end we will construct a 4-transitive permutation representation on 11 points. But first we consider the action of G on the design Γ.

Exercise 5.6. Use GAP to check the following assertions:

1. The group G is 3-transitive on the points of Γ.
2. G has an orbit of length 22 on the blocks.

Let $\mathcal{B}$ denote a G-orbit of length 22 on the set of blocks. Since G is 3-transitive on the 12 points of Γ, every triple of points is in a constant number λ of blocks in $\mathcal{B}$. An easy counting argument shows that $\lambda = 2$, and we have found that $(P, \mathcal{B})$ is a 3-(12,6,2) design.

For each block $b \in \mathcal{B}$, its complement $b' = \{1, \ldots, 12\} \setminus b$ is also a block in $\mathcal{B}$. The group G acts on the 11 pairs of complementary blocks of $\mathcal{B}$.

Exercise 5.7. Show that G induces a 4-transitive action on the 11 complementary block pairs of $\mathcal{B}$. Use this to prove that G is indeed the automorphism group of a 4-(11,5,1) design and conclude that it is isomorphic to M_{11}.

Exercise 5.8. If we fix the point 12 of the design $(P, \mathcal{B})$, and consider the residue of the design at that point, we obtain a 2-(11,5,2) design. Such a design is unique, see [6], and is called the *biplane of order* 11. Show that the full automorphism group of the biplane is isomorphic to the group $L_2(11)(= PSL_2(11))$ as described in Example 3.8 of Chapter 8. Check that the element d from Example 3.8 of Chapter 8 interchanges the points and blocks of the biplane.

Exercise 5.9. Use the permutation action of M_{12} on the 12 cosets of G to find that G is a 4-transitive group on 11 points. (This also reveals that G is isomorphic to M_{11}.)

5.10. 2-Transitive Subgroups of Order 660. One of the 2-transitive subgroups we have found in 5.4 is of order 660, which we now denote by H. We also encountered another group of order 660, namely the automorphism group of the biplane. Again, this is not a coincidence as will be shown in the sequel. For that purpose we consider the action of M_{12} on the 144 cosets of the subgroup H. A computation shows that H has two orbits of size 11 on the cosets in $H \setminus M_{12}$. Denote one of these two orbits by P and the other by B.

```
H:=Filtered(Twotrans,x->Size(x)=660)[1];
# now  H  is the 2-transitive subgroup of order  660.
M12:=Operation(m12,Cosets(m12,H),OnRight);
HH:=Stabilizer(M12,1);
Print("The orbits on [1..144] of the stabilizer HH of 1 are:\n",
 Orbits(HH,[1..144]),"\n");
Print("HH has size ",Size(HH),"\n");
```

Moreover, H is 2-transitive on P and on B. If we fix an element b in B, then the stabilizer in H of b has two orbits on P, one of length 5 and one of length 6. In this way we can associate to each $b \in B$ a subset P_b of size 5 in P. Denote by P_B the set of all eleven P_b.

Exercise 5.11. Check that (P, P_B) is a biplane of order 11 with automorphism group H.

Exercise 5.12. Conclude from the above computations that any proper 2-transitive subgroup of M_{12} is either isomorphic to M_{11} or to $L_2(11)$.

5.13. A Graph on Cosets. As we have seen before in 5.8, the group $L_2(11)$ admits an outer automorphism switching the points and blocks of the biplane. This automorphism can be extended to an outer automorphism of the group M_{12} switching the two classes of subgroups isomorphic to M_{11} we have encountered (point stabilizers in M_{12} and 2-transitive subgroups of order 7920). This automorphism can be found in the following way. On the 144 points of the permutation action of $G = M_{12}$ on the cosets of the group $H \simeq L_2(11)$ we define a graph structure Γ by calling two points x and y adjacent if and only if y is in one of the two orbits of size 11 under the action of the stabilizer G_x of x in G. If we fix a vertex x of Γ, then the 22 vertices adjacent to x can be identified with the points and blocks of the biplane of order 11. In fact, the subgraph induced on these 22 points is the incidence graph of the biplane. (Check this with the help of 5.11!) The automorphism group of this incidence graph of the biplane is the group $L_2(11){:}\,2$ and the group $M_{12}{:}\,2$ can be found as the automorphism group of the whole graph on 144 points.

In the GAP code below we us the GRAPE [19] share library package to construct the graph Γ, determine its automorphism group, check that Γ is connected, and study the induced subgraph on the neighbours of a vertex. The GRAPE package is used for computing with graphs and groups, and uses B. McKay's *nauty* [12] package for computing the automorphism group of a graph. More information on GRAPE functions can be found in the GAP manual.

```
RequirePackage("grape");
F:=Filtered(Orbits(HH,[1..144]),x->Length(x)=11);
# Now F[1][1] and F[2][1] are representatives of the two HH-orbits
# of length 11.
gamma:=EdgeOrbitsGraph(M12,[[1,F[1][1]],[1,F[2][1]]]);
AutM12:=AutGroupGraph(gamma);
Print("AutM12 has size ",Size(AutM12),"\n");
Print("gamma is connected is ",IsConnectedGraph(gamma),"\n");
delta:=DistanceSetInduced(gamma,[1],[1]);
# now  delta  is the induced subraph on the neighbours of
# the vertex 1.
Print("delta has order ",OrderGraph(delta),
 " and vertex-degree set ",VertexDegrees(delta),"\n");
Print("delta has (global) parameters ",
 GlobalParameters(delta),"\n");
```

6. Graphs Which Are Locally the Incidence Graph of the Biplane

Coset enumeration has many and varied applications in both group theory and geometry. For example, it has been used by many to study and construct group presentations (see [7, 16, 3]), to construct and characterize certain groups (including certain sporadic simple groups) (see [5, 13, 17] and the previous sections), and to classify certain finite geometries and graphs (see for example [1, 18, 20, 21, 22]). We shall give some of the flavour of these applications.

Let Γ and Δ be simple graphs (i.e., undirected, with no loops and no multiple edges). Then Γ is said to be *locally* Δ if for every vertex v of Γ, the induced subgraph on the neighbours of v is isomorphic to Δ. We shall use coset enumeration to study some presentations which arise in the classification of the connected, ordered-triangle-transitive graphs which are locally the incidence graph of the (unique) 2-(11,5,2) design. In the process, we obtain a presentation for $M_{12}{:}2$, the automorphism group of the sporadic simple Mathieu group M_{12}, which acts on a locally Δ graph on 144 points as we have seen in 5.13.

Let Δ be the incidence graph of the unique 2-(11,5,2) design $\mathcal{D}$ (see 5.8). Thus Δ has exactly 22 vertices: 11 corresponding to the points of $\mathcal{D}$ and 11 corresponding to the blocks of $\mathcal{D}$, with $\{v, w\}$ an edge of Δ precisely when $\{v, w\}$ is an incident point-block pair.

Let $H = \mathrm{Aut}(\Delta)$. Then $H \simeq L_2(11){:}2$, and, as was shown in Example 3.8 of Chapter 8, H can be presented as follows:

$$H = \langle a, b, c, d \mid a^3, b^2, c^2, d^2, (ab)^3, (ac)^2, a(cd)^4, (bc)^3, (bd)^2 \rangle.$$

In this presentation we find that $\langle a, b, c\rangle$ is the stabilizer of a vertex y, $\langle a, b\rangle$ is the pointwise stabilizer of an edge $\{y, z\}$ on that vertex, and $\langle a, b, d\rangle$ is the setwise stabilizer of the edge $\{y, z\}$; see also Example 3.8 of Chapter 8. We further remark that the relator $(ac)^2$ is unnecessary in the presentation for H. Indeed, $a = (cd)^{-4} = (dc)^4$, and so $(ac)^2 = (dc)^4c(dc)^4c = (dc)^4(cd)^4c^2 = 1$. In a similar way, we see that $(ad)^2 = 1$ holds in H.

If the connected graph Γ is locally Δ, and $\mathrm{Aut}(\Gamma)$ is transitive on the ordered-triangles $[u, v, w]$ of Γ, then it can be shown (see the Exercises 6.1 and 6.2 below) that the stabilizer in $\mathrm{Aut}(\Gamma)$ of a vertex v of Γ acts faithfully on the neighbourhood of v, and is isomorphic either to $L_2(11){:}\,2$ or its Sylow-11 normalizer $11{:}\,10$.

Exercise 6.1. (See also [18, Theorem 1] (due to Weetman).)

1. Prove that the only element of H fixing a point and each of the 5 blocks on this point is the identity element.
2. Let Γ be a connected graph which is locally Δ. Suppose $g \in \mathrm{Aut}(\Gamma)$ fixes a vertex v and all its neighbours. Let w be a vertex adjacent to v. Show that g fixes the 5 common neighbours of v and w, and conclude that g fixes all neighbours of w.
3. Prove that $g = 1$.

Exercise 6.2. Let K be a subgroup of $H \simeq L_2(11){:}\,2$ that is transitive on the ordered pairs of adjacent vertices of the incidence graph Δ of the biplane. Prove that K is either a Sylow-11 normalizer $11{:}\,10$ in H, or is H itself.

We now consider the case where Γ is a connected, ordered-triangle-transitive, locally Δ graph, with vertex stabilizer $L_2(11){:}\,2$ (see below) in $\mathrm{Aut}(\Gamma)$.

Suppose such a Γ exists, and let $\{x, y, z\}$ be a triangle of Γ. Then $G = \mathrm{Aut}(\Gamma)$ is a quotient of the group formed by amalgamating the G-stabilizers X, Y, Z of x, $\{x, y\}$, $\{x, y, z\}$, respectively. Analysis of Δ and $L_2(11){:}\,2$ shows that X, Y, Z must be respectively isomorphic to

$$L_2(11){:}\,2, \quad \mathrm{S}_5, \quad (\mathrm{A}_4 \times 3){:}\,2.$$

Furthermore, we must have $X \cap Y \simeq \mathrm{A}_5$, $X \cap Z \simeq \mathrm{S}_4$, and $Y \cap Z \simeq \mathrm{S}_4$. In fact, we will show that the group G can be generated by elements a, b, c, d and e, such that the following relators hold:

$$a^3, b^2, c^2, d^2, (ab)^3, a(cd)^4, (bc)^3, (bd)^2, e^2, (ae)^2, (be)^2, (ce)^2, (de)^3. \tag{6.1}$$

(Recall that $1 = (ac)^2 = (ad)^2$ are consequences of these relators.)

Indeed, if we fix a vertex x, then the stabilizer X of x is isomorphic to $L_2(11){:}\,2$ and can be generated by elements a, b, c and d such that the relators above which do not involve e are satisfied. Furthermore, $\langle a, b, c\rangle \simeq \mathrm{A}_5$ is the stabilizer in X of some vertex y adjacent to x, and $\langle a, b\rangle \simeq \mathrm{A}_4$ is the pointwise stabilizer in X of an edge $\{y, z\}$ in the induced subgraph on the neighbours of x.

Let $[x, y, z]$ denote the ordered-triangle containing the above vertices x, y, z (in that order). Since we are assuming that G acts transitively on the ordered-triangles of Γ, there is an element $e \in G$ such that $[x, y, z]^e = [y, x, z]$. Then $\langle a, b, c, e\rangle$ is of shape $\mathrm{A}_5.2$, and $\langle a, b, e\rangle$ is of shape $\mathrm{A}_4.2$. Since we have $\langle a, b, e\rangle \leq G_z \simeq L_2(11){:}\,2$, we conclude (from the subgroup structure of $L_2(11){:}\,2$ (see [2])) that $\langle a, b, e\rangle \simeq \mathrm{S}_4$, and so we must have $\langle a, b, c, e\rangle \simeq \mathrm{S}_5$. Moreover, we can choose the element e in this S_5 such that the following relations hold:

$$e^2 = (ae)^2 = (be)^2 = (ce)^2 = 1.$$

(Compare this with the relations involving d.) Since Γ is assumed to be connected, the group G is generated by a, b, c, d and e. It remains to check that the relation $(de)^3 = 1$ holds. Since $[x, y, z]^d = [x, z, y]$ and $[x, y, z]^e = [y, x, z]$, the element $(de)^3$ fixes each of the three vertices x, y and z of Γ and therefore is contained in $\langle a, b\rangle$. On the other hand we already know that $ade = da^{-1}e = dea$ and $bde = dbe = deb$. So $(de)^3 \in Z(\langle a, b\rangle) = 1$.

Now let $\tilde{G}$ be the group presented by generators a, b, c, d and e subject (only) to the relators given in (6.1). We shall determine the graphs Γ by determining each homomorphic image G of $\tilde{G}$, such that $\langle a, b, c, d\rangle$ maps (isomorphically) onto $X \simeq L_2(11){:}\,2$, $\langle a, b, c, e\rangle$ maps (isomorphically) onto $Y \simeq \mathrm{S}_5$ and $\langle a, b, d, e\rangle$ maps (isomorphically) onto $Z \simeq (\mathrm{A}_4 \times 3){:}\,2$. Such homomorphic images G of $\tilde{G}$ are the candidates for the automorphism groups of the graphs we seek, such that $X, Y, Z \leq G$ would be the respective stabilizers of a vertex, an edge on that vertex, and a triangle on that edge. Given a candidate G, and X and Y, we construct the corresponding candidate graph Γ as follows. The vertices are the right cosets of X in G, and the edge-set is the G-orbit of $\{X, Xt\}$, where t is any element of $Y \setminus X$. We can then check whether Γ is locally a graph of order 22 and degree 5. If so, it then follows from our construction that the G-stabilizer of a vertex of Γ is $L_2(11){:}\,2$, Γ is connected, G acts transitively on the ordered-triangles of Γ, and Γ is locally Δ.

Exercise 6.3. Prove the assertions of the previous sentence.

Exercise 6.4. Apply coset enumeration, and find that $\langle a, b, c, d\rangle$ has index 432 in $\tilde{G}$. What is the order of $\tilde{G}$?

Exercise 6.5. Using the presentation (6.1) and coset enumeration, calculate generators for the degree 432 permutation group formed by $\tilde{G}$ acting (faithfully) on the cosets of $X = \langle a, b, c, d\rangle$. Calculate the orbits of X. Consider the graph Γ_{432} whose edge-set is $\{x, x^e\}^{\tilde{G}}$, where x is a vertex stabilized by X. Show that Γ is locally the incidence graph Δ of the biplane of order /11. (You might try to use some appropriate GRAPE functions.)

As we have seen in 5.13, the group $M_{12}{:}2$ is the automorphism group of a connected graph Γ on 144 points which is also locally Δ, and has vertex stabilizer $L_2(11){:}2$. It is easy to see that Γ is ordered-triangle-transitive. It follows that $M_{12}{:}2$ is a quotient of the group $\tilde{G}$. Since the order of $\tilde{G}$ is 3 times the order of $M_{12}{:}2$, the group $\tilde{G}$ contains a normal subgroup of order 3.

Exercise 6.6. Show with the help of coset enumeration that the subgroup $N = \langle (bcde)^{11} \rangle$ is a normal subgroup of order 3 in $\tilde{G}$.

What are the orbits of $\langle (bcde)^{11} \rangle$ on the 432 points of the graph Γ_{432}?

Construct a graph Γ_{144}, with as vertices the 144 orbits of $\langle (bcde)^{11} \rangle$ on the vertices of Γ_{432}, such that Γ_{144} is also locally Δ.

It follows from 5.13 and the above that $\mathrm{Aut}(\Gamma_{144}) \simeq M_{12}{:}2$,

$$\tilde{G}/\langle (bcde)^{11} \rangle \simeq M_{12}{:}2,$$

and

$$\tilde{G} \simeq (3 \times M_{12}){:}2$$

(note that $\tilde{G}$ has the symmetric group S_3 as a homomorphic image). We conclude that Γ_{144} and Γ_{432} are the only connected, ordered-triangle-transitive, locally Δ graphs whose vertex stabilizer is $L_2(11){:}2$.

Note that we obtain a presentation for $M_{12}{:}2$ from the presentation (6.1) by adjoining the relator $(bcde)^{11}$.

Exercise 6.7. Give an isomorphism between the group $\tilde{G}/N$ and the group $M_{12}{:}2$ as given in 5.13.

Exercise 6.8. (Challenging) As usual, let Δ be the incidence graph of the biplane $\mathcal{D}$. Show that if G is the automorphism group of a connected, ordered-triangle-transitive, locally Δ graph Γ, such that the G-stabilizer of a vertex is $11{:}10$, then G is a quotient of

$$\tilde{G} = \langle a, b, c \mid a^5, b^2, c^2, (ac)^2, (bc)^3, [a,b]^{11}, [b,a]^a[a,b]^3 \rangle,$$

such that for some triangle $\{x, y, z\}$ of Γ, the G-stabilizers of x, $\{x, y\}$, $\{x, y, z\}$ are the respective images of $\langle a, b \rangle$, $\langle a, c \rangle$, $\langle b, c \rangle$. (In particular, show that replacing the relator $(ac)^2$ by $[a, c]$ in the presentation above does not lead to such a graph.)

Study the permutation group $\tilde{G}$ acting on the cosets of $\langle a, b \rangle$, using the permutation group algorithms in GAP or MAGMA. Determine the order and structure of $\tilde{G}$.

Prove that $\tilde{G}$ is the automorphism group of an ordered-triangle-transitive, connected, locally Δ graph, such that the $\tilde{G}$-stabilizer of a vertex is $11{:}10$.

Prove that, up to isomorphism, there is just one ordered-triangle-transitive, connected, locally Δ graph Γ, such that the $\mathrm{Aut}(\Gamma)$-stabilizer of a vertex is $11{:}10$.

References

1. J. van Bon (1993): *Some extended generalized hexagons*, pp. 395–403 in Finite Geometry and Combinatorics (F. De Clerck et al., eds), LMS Lecture Notes **191**, Cambridge University Press, Cambridge.
2. J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson (1985): *ATLAS of Finite Groups*, Clarendon Press, Oxford.
3. J. H. Conway, S. P. Norton, and L. H. Soicher (1988): The Bimonster, the group Y_{555}, and the projective plane of order 3, pp. 27–50 in Computers in Algebra (M.C. Tangora, ed.), Marcel Dekker, New York.
4. J. H. Conway and N. J. A. Sloane (1988): *Sphere Packings, Lattices and Groups*, Springer-Verlag, Berlin Heidelberg New York.
5. H. Cuypers and J. I. Hall (1992): *The classification of 3-transposition groups with trivial centre*, pp. 121–138 in Groups, Combinatorics and Geometry (M.W. Liebeck and J. Saxl, eds), LMS Lecture Notes **165**, Cambridge University Press.
6. D. R. Hughes and F. C. Piper (1985): *Design Theory*, Cambridge University Press, Cambridge (reprinted in paperback, 1988).
7. D. L. Johnson (1990): *Presentations of Groups*, Cambridge University Press, Cambridge.
8. H. Lüneburg (1969): *Transitieve Erweiterungen endlicher Permutationsgruppen*, Lecture Notes in Math. **84**, Springer-Verlag, Berlin Heidelberg New York.
9. E. Mathieu (1860): *Mémoire sur le nombre de valeurs que peut acquérir une function quand on y permut ses variables de toutes le manière possibles*, J. de Math. Pure et App. **5**, 9–42.
10. E. Mathieu (1861): *Mémoire sur l'étude des functions de plusieures quantités, sur la manière des formes et sur les substitutions qui laissent invariables*, J. de Math. Pure et App. **6**, 241–323.
11. E. Mathieu (1873): *Sur la function cinq fois transitive des 24 quantités*, J. de Math. Pure et App. **18**, 25–46.
12. B. D. McKay (1990): *nauty user's guide (version 1.5)*, Technical report TR-CS-90-02, Computer Science Department, Australian National University.
13. J. McKay (1974): *Computing with finite simple groups*, pp. 448–452 in Proceedings, Second International Conference on the Theory of Groups, Canberra, 1973, Lecture Notes in Mathematics **372**, Springer-Verlag, New York Heidelberg Berlin.
14. C. E. Praeger and L. H. Soicher (1997): *Low Rank Representations and Graphs for Sporadic Groups*, Australian Math. Soc. Lecture Series **8**, Cambridge University Press, Cambridge.
15. M. Schönert, et al. (1994): GAP – *Groups, Algorithms and Programming, version 3, release 4*, Lehrstuhl D für Mathematik, RWTH Aachen.
16. L. H. Soicher (1988): *Presentations for some groups related to* Co_1, pp. 151–154 in Computers in Algebra (M.C. Tangora, ed.), Marcel Dekker, New York.
17. L. H. Soicher (1990): *A new existence and uniqueness proof for the O'Nan group*, Bull. London Math. Soc. **22**, 148–152.
18. L. H. Soicher (1992): *On simplicial complexes related to the Suzuki sequence graphs*, pp. 240–248 in Groups, Combinatorics and Geometry (M.W. Liebeck and J. Saxl, eds), LMS Lecture Notes **165**, Cambridge University Press, Cambridge.
19. L. H. Soicher (1993): GRAPE*: a system for computing with graphs and groups*, pp. 287–291 in Groups and Computation, (L. Finkelstein and W.M. Kantor, eds), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **11**, Amer. Math. Soc.

20. R. Weiss (1990): *Extended generalized hexagons*, Math. Proc. Camb. Phil. Soc. **108**, 7–19.
21. R. Weiss (1991): *A geometric characterization of the groups* McL *and* Co_3, J. London Math. Soc. **44**, 261–269.
22. S. Yoshiara (1991): *A classification of flag-transitive classical* c.C_2*-geometries by means of generators and relations*, Europ. J. Combinatorics **12**, 159–181.

Project 7: The Golay Codes

Mario de Boer and Ruud Pellikaan

1. Introduction

In this project we give examples of methods described in the Chapters 10 and 11 on finding the minimum weight codewords, the decoding of cyclic codes and working with the Mathieu groups (see also 6). The codes that we use here are the well-known *Golay codes.* These codes are among the most beautiful objects in coding theory, and we would like to give some reasons why.

There are two Golay codes: the ternary cyclic code $\mathcal{G}_{11}$ and the binary cyclic code $\mathcal{G}_{23}$. The ternary Golay code $\mathcal{G}_{11}$ has parameters $[11, 6, 5]$, and it is the unique code with these parameters. The automorphism group $\mathrm{Aut}(\mathcal{G}_{11})$ is the Mathieu group M_{11}. The group M_{11} is simple, 4-fold transitive and has size $11 \cdot 10 \cdot 9 \cdot 8$. The supports of the codewords of weight 5 form the blocks of a 4-design, the unique Steiner system $S(4, 5, 11)$. The ternary Golay code is a perfect code; this means that the Hamming spheres of radius $(d-1)/2 = 2$ centered at the codewords of $\mathcal{G}_{11}$ exactly cover the whole space $\mathbb{F}_3^{11}$. The code $\mathcal{G}_{11}$ can be uniquely extended to a $[12, 6, 6]$ code, which we will denote by $\mathcal{G}_{12}$. The code $\mathcal{G}_{12}$ is self-dual and $\mathrm{Aut}(\mathcal{G}_{12}) = M_{12}$: the simple, 5-fold transitive Mathieu group of size $12 \cdot 11 \cdot 10 \cdot 9 \cdot 8$. The supports of the codewords of weight 6 in $\mathcal{G}_{12}$ form a 5-design, the unique $S(5, 6, 12)$.

The binary Golay code $\mathcal{G}_{23}$ has similar properties. Its parameters are $[23, 12, 7]$, and it is the unique code with these parameters. The automorphism group $\mathrm{Aut}(\mathcal{G}_{23})$ is the Mathieu group M_{23}. The group M_{23} is simple, 4-fold transitive and has size $23 \cdot 22 \cdot 21 \cdot 20 \cdot 48$. The supports of the codewords of weight 7 form the blocks of a 4-design, the unique Steiner system $S(4, 7, 23)$. The binary Golay code is a perfect code, so the Hamming spheres of radius 3 centered at the codewords of $\mathcal{G}_{11}$ exactly cover the whole space $\mathbb{F}_2^{23}$. The code $\mathcal{G}_{23}$ can be uniquely extended to a $[24, 12, 8]$ code, which we will denote by $\mathcal{G}_{24}$. The code $\mathcal{G}_{24}$ is self-dual and $\mathrm{Aut}(\mathcal{G}_{24}) = M_{24}$: the simple, 5-fold transitive Mathieu group of size $24 \cdot 23 \cdot 22 \cdot 21 \cdot 20 \cdot 48$. The supports of the codewords of weight 8 in $\mathcal{G}_{24}$ form a 5-design, the unique $S(5, 8, 24)$.

2. Minimal Weight Codewords of $\mathcal{G}_{11}$

$\mathcal{G}_{11}$ is the ternary cyclic code of length 11 with defining set $J = \{1\}$. It is a $[11, 6, d]$ code with complete defining set $J(\mathcal{G}_{11}) = \{1, 3, 4, 5, 9\}$. The generator polynomial is

$$g(X) = \prod_{j \in J(\mathcal{G}_{11})} (X - \alpha^j) = 2 + X^2 + 2X^3 + X^4 + X^5.$$

From the BCH bound we see that $d \geq 4$, and by computing Gröbner bases we will show that in fact $d = 5$. Moreover, we will determine all codewords of minimal weight.

First we consider the system $\mathcal{S}_{\mathcal{G}_{11}}(4)$:

$$\mathcal{S}_{\mathcal{G}_{11}}(4) = \left\{ \begin{array}{lcl} A_5 + \sigma_1 A_4 + \sigma_2 A_3 + \sigma_3 A_2 + \sigma_4 A_1 & = & 0 \\ A_6 + \sigma_1 A_5 + \sigma_2 A_4 + \sigma_3 A_3 + \sigma_4 A_2 & = & 0 \\ \vdots & \vdots & \vdots \\ A_4 + \sigma_1 A_3 + \sigma_2 A_2 + \sigma_3 A_1 + \sigma_4 A_0 & = & 0 \\ \quad A_j = 0 \ \text{ for } \ j \in J(\mathcal{G}_{11}) & & \\ \quad A_{3j} = A_j^3 \ \text{ for } \ j = 1, \ldots, 11. & & \end{array} \right.$$

Using $A_{3i} = A_i^3$ we can express every A_i with $i \in \{1, 2, \ldots, 10\} \setminus J(\mathcal{G}_{11})$ as a power of A_2 (this can be done since all of these i form a single cyclotomic coset). Setting $A_i = 0$ for $i \in J(\mathcal{G}_{11})$ and writing $A_2 = a$ and $A_0 = b$ this reduces $\mathcal{S}_{\mathcal{G}_{11}}(4)$ to

$$\mathcal{S}_{\mathcal{G}_{11}}(4) = \left\{ \begin{array}{lcl} \sigma_3 a & = & 0 \\ a^3 + \sigma_4 a & = & 0 \\ a^9 + \sigma_1 a^3 & = & 0 \\ a^{81} + \sigma_1 a^9 + \sigma_2 a^3 & = & 0 \\ \sigma_1 a^{81} + \sigma_2 a^9 + \sigma_3 a^3 & = & 0 \\ a^{27} + \sigma_2 a^{81} + \sigma_3 a^9 + \sigma_4 a^3 & = & 0 \\ b + \sigma_1 a^{27} + \sigma_3 a^{81} + \sigma_4 a^9 & = & 0 \\ \sigma_1 b + \sigma_2 a^{27} + \sigma_4 a^{81} & = & 0 \\ a + \sigma_2 b + \sigma_3 a^{27} & = & 0 \\ \sigma_1 a + \sigma_3 b + \sigma_4 a^{27} & = & 0 \\ \sigma_2 a + \sigma_4 b & = & 0 \\ b^3 - b & = & 0. \end{array} \right.$$

Computing a Gröbner basis $\mathcal{G}$ with respect to the lexicographic order with

$$\sigma_4 > \sigma_3 > \sigma_2 > \sigma_1 > b > a$$

gives $\mathcal{G} = \{b, a\}$ and hence there are no nonzero codewords of weight at most 4. We conclude $d \geq 5$, and even $d = 5$, since the weight of the generator polynomial is $\text{wt}(g(X)) = 5$. To determine the minimum weight codewords we consider the system $\mathcal{S}_{\mathcal{G}_{11}}(5)$:

$$\mathcal{S}_{\mathcal{G}_{11}}(5) = \left\{ \begin{array}{lcl} A_6 + \sigma_1 A_5 + \sigma_2 A_4 + \sigma_3 A_3 + \sigma_4 A_2 + \sigma_5 A_1 & = & 0 \\ A_7 + \sigma_1 A_6 + \sigma_2 A_5 + \sigma_3 A_4 + \sigma_4 A_3 + \sigma_5 A_2 & = & 0 \\ \vdots & \vdots & \vdots \\ A_5 + \sigma_1 A_4 + \sigma_2 A_3 + \sigma_3 A_2 + \sigma_4 A_1 + \sigma_5 A_0 & = & 0 \\ \quad A_i = 0 \ \text{ for } \ i \in J(\mathcal{G}_{11}) & & \\ \quad A_{3i} = A_i^3 \ \text{ for } \ i = 0, \ldots, 10 & & . \end{array} \right.$$

Again we can reduce the system as we did in the system $\mathcal{S}_{\mathcal{G}_{11}}(4)$ and compute its Gröbner basis with respect to the lexicographic order with

$$\sigma_5 > \sigma_4 > \sigma_3 > \sigma_2 > \sigma_1 > b > a.$$

The resulting basis $\mathcal{G}$ is (after 2 minutes using Axiom or 10 minutes using Macaulay)

$$\mathcal{G} = \begin{cases} \sigma_5 a + 2a^{31} + 2a^9, \sigma_4 a + a^3, \sigma_3 a + 2a^{107} + a^{41} + 2a^{19}, \\ \sigma_2 a + a^{79} + 2a^{35} + a^{13}, \sigma_1 a + a^{29} + 2a^7, \\ b + a^{77} + 2a^{55} + a^{33} + a^{11}, a^{133} + 2a^{111} + 2a^{89} + 2a^{67} + a^{45} + a, \end{cases}$$

where $a = A_2$ and $b = A_0$. From the triangular form of the basis $\mathcal{G}$, it is easy to see that the number of codewords of weight 5 in $\mathcal{G}_{11}$ equals the number of nonzero solutions to

$$f(X) = X^{133} + 2X^{111} + 2X^{89} + 2X^{67} + X^{45} + X = 0$$

in $\mathbb{F}_{3^5}$. We determine these solutions in the following exercise.

Exercise 2.1. Let $\alpha \in \mathbb{F}_{3^5}$ be a primitive element. Now show
1. $f(1) = 0$;
2. $f(\alpha^2) = 0$ (you can use a computer algebra package for this);
3. $f(\alpha^{11}X) = \alpha^{11} f(X)$.

Conclude from this that the complete set of zeros of $f(X)$ in $\mathbb{F}_{3^5} \setminus \{0\}$ is

$$M = \{\alpha^{i+11j} \mid i \in \{0,1,\ldots,10\} \setminus J(\mathcal{G}_{11}),\ j \in \{0,1,\ldots,21\}\}.$$

So the number of codewords of weight 5 is $\#M = 132$ and the locators of these words (i.e., the polynomials having as zeros the reciprocals of positions where the codewords have a nonzero value) are given by

$$\sigma(X,a) = \begin{cases} (a^{30} + a^8)X^5 + 2a^2X^4 + (a^{106} + 2a^{40} + a^{18})X^3 + \\ (2a^{78} + a^{34} + 2a^{12})X^2 + (2a^{28} + a^6)X + 1, \end{cases}$$

with $a \in M$.

Since the code is cyclic, any shift of a codeword of weight 5 is again a codeword of weight 5. We can recognize this fact from M in the following way.

Exercise 2.2. Show that there exists a primitive 11-th root of unity β such that $\sigma(X, \alpha^{11}a) = \sigma(\beta X, a)$ for all $a \in M$.

Now we can conclude that the codewords of weight 5 consist of the 6 codewords with locator polynomials $\sigma(X,a)$, $a \in \{1, \alpha^2, \alpha^6, \alpha^7, \alpha^8, \alpha^{10}\}$, their cyclic shifts, and their nonzero multiples in $\mathbb{F}_3^{11}$.

Exercise 2.3. Let α again be a primitive element in $\mathbb{F}_{3^5}$, then $\beta = \alpha^{22}$ is a fixed 11-th root of unity. Check that the zeros of the 6 polynomials $\sigma(X,a)$ are:

polynomial	$\{i \mid \beta^{-i}$ is a zero$\}$
$\sigma(X,1)$	2, 6, 7, 8, 10
$\sigma(X,\alpha^2)$	3, 4, 9, 10, 11
$\sigma(X,\alpha^6)$	1, 5, 8, 9, 11
$\sigma(X,\alpha^7)$	1, 2, 8, 10, 11
$\sigma(X,\alpha^8)$	2, 3, 5, 7, 9
$\sigma(X,\alpha^{10})$	3, 5, 8, 10, 11

Let $\mathcal{B}$ consist of the 6 subsets of $\{1,\dots,11\}$ in the table and their cyclic shifts modulo 11. Then $|\mathcal{B}| = 66$. Show that $\mathcal{B}$ is the set of blocks of a 4-design, the Steiner system $S(4,5,11)$.

3. Decoding of $\mathcal{G}_{23}$ with Gröbner Bases

Let $\mathcal{G}_{23}$ be the binary cyclic code of length 23 with defining set $J = \{1\}$. Then the complete defining set is $J(\mathcal{G}_{23}) = \{1,2,3,4,6,8,9,12,13,16,18\}$ and the code has parameters $[23,12,d]$. The BCH bound states that $d \geq 5$ but in fact $d = 7$. This can be checked in the same way as we did in the previous section for the ternary Golay code. The computer algebra packages we tried, did not perform very well on the systems $S_{\mathcal{G}_{23}}(w)$. Since the minimum distance is 7, $\mathcal{G}_{23}$ should be able to correct errors of weight at most 3. In this example we will decode a word with three errors.

Take

$$\mathbb{F}_{2^{11}} = \mathbb{F}_2[\beta]/(\beta^{11}+\beta^2+1)$$

and set $\alpha = \beta^{89}$. Then β is a primitive element of $\mathbb{F}_{2^{11}}$ and α has order 23.

The generator polynomial of the code is

$$g(X) = \prod_{j \in J(\mathcal{G}_{23})} (X - \alpha^j) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}.$$

Suppose we send the codeword $g(X)$, which corresponds to the binary vector

$$\mathbf{c} = (1,1,0,0,0,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0)$$

over a noisy channel, and the following error occurs during transmission:

$$\mathbf{e} = (1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0).$$

As a result at the other end of the channel the following vector will be received:

$$\mathbf{y} = (0,1,0,1,0,1,1,1,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0),$$

corresponding to the polynomial

$$r(X) = X + X^3 + X^5 + X^6 + X^7 + X^9 + X^{11} + X^{17}.$$

We will now decode the received word by applying the decoding algorithm.

First we compute the syndrome:

$$s_1 = H\mathbf{y} = r(\alpha) = \alpha+\alpha^3+\alpha^5+\alpha^6+\alpha^7+\alpha^9+\alpha^{11}+\alpha^{17} = \beta^9+\beta^6+\beta^3+\beta^2+1.$$

Since $s_1 \neq 0$ we see that errors have occurred during transmission.

We already remarked that the Y_i variables can be disposed of by setting them equal to 1, since 1 is the only error value that can occur.

Following the algorithm of Section 3 of Chapter 11 we set

$$\mathcal{S} = \{X_1 + \beta^9 + \beta^6 + \beta^3 + \beta^2 + 1, X_1^{23} + 1\}$$

and can conclude that there are no solutions since s_1 is not a 23-rd root of unity.

In the next step we set

$$\mathcal{S} = \{X_2 + X_1 + \beta^9 + \beta^6 + \beta^3 + \beta^2 + 1, X_2^{23} + 1, X_1^{23} + 1\}$$

and compute its Gröbner basis with respect to the lexicographic order with $X_2 > X_1$:

$$\mathcal{G} = \{1\}.$$

Since $1 \in \mathcal{G}$ there is no solution to these syndrome equations and we proceed with the loop of the algorithm. We set

$$\mathcal{S} = \{X_3 + X_2 + X_1 + \beta^9 + \beta^6 + \beta^3 + \beta^2 + 1, X_3^{23} + 1, X_2^{23} + 1, X_1^{23} + 1\},$$

and a Gröbner basis with respect to the lexicographic order with $X_3 > X_2 > X_1$ is computed:

$$\begin{cases} X_3 + X_2 + X_1 + \beta^9 + \beta^6 + \beta^3 + \beta^2 + 1, \\ X_2^2 + X_2X_1 + (\beta^9 + \beta^6 + \beta^3 + \beta^2 + 1)X_2 + X_1^2 + \\ \qquad + (\beta^9 + \beta^6 + \beta^3 + \beta^2 + 1)X_1 + \beta^6 + \beta^5 + \beta^2, \\ X_1^3 + (\beta^9 + \beta^6 + \beta^3 + \beta^2 + 1)X_1^2 + (\beta^6 + \beta^5 + \beta^2)X_1 + \beta^9 + \beta^5 + \beta^3. \end{cases}$$

This took 8 minutes using Axiom. We did the same computation with $X_j^{24} + X_j$ instead of $X_j^{23} + 1$ for $j = 1, 2, 3$ and it took only 90 seconds.

Now $1 \notin \mathcal{G}$ and there are solutions to the syndrome equations. The error-locator polynomial is

$$g(X_1) = X_1^3 + (\beta^9 + \beta^6 + \beta^3 + \beta^2 + 1)X_1^2 + (\beta^6 + \beta^5 + \beta^2)X_1 + \beta^9 + \beta^5 + \beta^3$$

and its zeros are the error-locators $\{\alpha^0, \alpha^3, \alpha^{17}\}$. Hence the errors occurred at positions 0, 3 and 17 and the word that was sent is

$$\mathbf{y} - (1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0) =$$

$$(1,1,0,0,0,1,1,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0).$$

We have recovered the transmitted codeword c.

4. One-Step Decoding of $\mathcal{G}_{23}$

In this paragraph we will decode all error patterns of weight 3 that can occur in a codeword of the code $\mathcal{G}_{23}$ at once by computing the Gröbner basis for variable syndromes S. Apart from the advantage that all syndromes are treated at once, it also has the advantage that the computations take place over the field $\mathbb{F}_2$ instead of the large field $\mathbb{F}_{2^{11}}$. The system of equations is:

$$\mathcal{S} = \left\{ \begin{array}{ccc} X_3 + X_2 + X_1 + S & = & 0 \\ X_3^{23} + 1 & = & 0 \\ X_2^{23} + 1 & = & 0 \\ X_1^{23} + 1 & = & 0. \end{array} \right.$$

The outcome of this set of equations is quite complicated. The result is much simpler if we consider the following set of equations.

$$\mathcal{S}' = \left\{ \begin{array}{ccc} X_3 + X_2 + S + X_1 & = & 0 \\ X_3^{24} + X_3 & = & 0 \\ X_2^{24} + X_2 & = & 0 \\ X_1^{24} + X_1 & = & 0. \end{array} \right.$$

With the lexicographic order with $X_3 > X_2 > X_1 > S$, the computer was still not finished with its computations after 24 hours. Loustaunau and York did this example where they started with the above system, which is a Gröbner basis with respect to the lexicographic order with $S > X_3 > X_2 > X_1$, and transformed it into a Gröbner basis with respect to the lexicographic order with $X_3 > X_2 > X_1 > S$ as explained in the notes of Chapter 11. Using the lexicographic order with $X_3 > X_2 > S > X_1$ we obtain the Gröbner basis:

$$\mathcal{G} = \left\{ \begin{array}{l} X_3 + X_2 + S + X_1, \\ X_2^{24} + X_2, \\ X_2^2 S + X_2^2 X_1 + X_2 S^2 + X_2 X_1^2 + S^{256} + S^3 + S^2 X_1 + S X_1^2, \\ g(X_1), \\ X_1^{24} + X_1, \end{array} \right.$$

with

$$g(X_1) = \left\{ \begin{array}{lcl} (S^{256} + S^3)X_1^{21} & + & (S^{257} + S^4)X_1^{20} + \\ (S^{260} + S^7)X_1^{17} & + & (S^{261} + S^8)X_1^{16} + \\ (S^{32} + S^9)X_1^{15} & + & (S^{33} + S^{10})X_1^{14} + \\ (S^{34} + S^{11})X_1^{13} & + & (S^{35} + S^{12})X_1^{12} + \\ (S^{36} + S^{13})X_1^{11} & + & (S^{37} + S^{14})X_1^{10} + \\ (S^{38} + S^{15})X_1^{9} & + & (S^{39} + S^{16})X_1^{8} + \\ (S^{40} + S^{17})X_1^{7} & + & (S^{64} + S^{41})X_1^{6} + \\ (S^{272} + S^{42})X_1^{5} & + & (S^{273} + S^{66} + S^{43} + S^{20})X_1^{4} + \\ (S^{44} + S^{21})X_1^{3} & + & (S^{68} + S^{45})X_1^{2} + \\ (S^{276} + S^{46})X_1 & + & (S^{277} + S^{70} + S^{47} + S). \end{array} \right.$$

We conclude that for a general syndrome S we find the error-locator polynomial

$$\gcd(g(X_1), X_1^{23}+1).$$

These computations took 120 seconds using Axiom. The original set of equations S took 150 seconds. Macaulay did both these computations on the same computer in 3 seconds.

Exercise 4.1. Check that the coefficient of X^i is divisible by $S^{23}+1$ for all i.

Exercise 4.2. Suppose $s = x_1 + x_2 + x_3$ with $x_j \in \mathbb{F}_{2^{11}}$ and $x_j^{23} = 1$ for all j. Show that $s^{23} = 1$ if and only if $x_i = x_j$ for some i, j with $1 \leq i < j \leq 3$.

Exercise 4.3. Denote $g(X_1)/(S^{23}+1)$ by $h(X_1)$. Compute $\gcd(h(X_1), X_1^{23}+1)$ with Euclid's algorithm in the ring $\mathbb{F}_q(S)[X_1]$ and show that it is a polynomial of degree 3 in X_1 with rational functions in S as coefficients.

5. The Key Equation for $\mathcal{G}_{23}$

In this section we will use the Euclidean algorithm to decode an error that occurred during the transmission of a codeword of the binary Golay code $\mathcal{G}_{23}$. As we mentioned in Section 4 of Chapter 11, decoding a cyclic code C by solving the key equation only works for errors of weight at most $(\delta-1)/2$, where δ is maximal such that $\{1, 2, \ldots, \delta - 1\} \subset J(C)$. In the case of the binary Golay code, this means we can only expect to decode errors of weight at most 2 in this way.

As in the previous section, we assume that the transmitted codeword was $g(X)$. Suppose the following error occurs:

$$\mathbf{e} = (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0).$$

Then the received word is

$$\mathbf{y} = (0,1,0,0,0,1,1,1,0,1,0,1,0,0,0,0,0,1,0,0,0,0,0),$$

corresponding to the polynomial

$$r(X) = X + X^5 + X^6 + X^7 + X^9 + X^{11} + X^{17}.$$

After we receive this word, we can compute the following syndromes:

$$\begin{array}{lllllll}
s_1 & = & r(\alpha) & & & = & \beta^{10}+\beta^9+\beta^7+\beta^6+1 \\
s_2 & = & r(\alpha^2) & = & s_1^2 & = & \beta^7+\beta^5+\beta^2+\beta \\
s_3 & = & r(\alpha^3) & = & s_1^{256} & = & \beta^8+\beta^7+\beta^6+\beta^5 \\
s_4 & = & r(\alpha^4) & = & s_1^4 & = & \beta^{10}+\beta^5+\beta^4+\beta^3+\beta^2.
\end{array}$$

Following Section 4 of Chapter 11 we define

$$S(Z) = s_1 + s_2 Z + s_3 Z^2 + s_4 Z^3$$

and we start the Euclidean algorithm on $S(Z)$ and Z^4. We find

$$Z^4 = S(Z) q_1(Z) + r_1(Z),$$

with

$$q_1(Z) = (\beta^9 + \beta^3 + \beta^2 + 1)Z + \beta^{10} + \beta^9 + \beta^5 + \beta$$

and

$$\begin{aligned} r_1(Z) \quad = \quad & (\beta^{10} + \beta^9 + \beta^7 + \beta^6 + \beta^5 + \beta^4)Z^2 + \\ & (\beta^{10} + \beta^9 + \beta^7 + \beta^5 + \beta^4 + \beta^3)Z + \\ & (\beta^9 + \beta^6 + \beta^2 + 1). \end{aligned}$$

In the following step we get

$$S(Z) = r_1(Z) q_2(Z) + r_2(Z),$$

with

$$q_2(Z) = (\beta^{10} + \beta^3 + \beta^2 + 1)Z + (\beta^{10} + \beta^7 + \beta^6 + \beta)$$

and

$$r_2(Z) = \beta^7 + \beta^6 + \beta^3 + \beta^2 + \beta + 1.$$

Since $\deg(r_1(Z)) \geq 2$ and $\deg(r_2(Z)) \leq 1$ we can stop the algorithm and compute

$$\begin{aligned} U_2(Z) \quad &= \quad q_2(Z) U_1(Z) + U_0(Z) \\ &= \quad q_2(Z) q_1(Z) + 1 \\ &= \quad (\beta^9 + \beta^8 + \beta^6)Z^2 + \\ & \quad\quad (\beta^7 + \beta^6 + \beta^3 + \beta^2 + \beta + 1)Z + \\ & \quad\quad \beta^9 + \beta^8 + \beta^7 + \beta^3 + \beta^2 + \beta + 1. \end{aligned}$$

From this we find

$$\sigma(Z) \quad = \quad U_2(Z)/(\beta^9 + \beta^8 + \beta^7 + \beta^3 + \beta^2 + \beta + 1) =$$

$$(\beta^{10} + \beta^9 + \beta^7 + \beta^6)Z^2 + (\beta^{10} + \beta^9 + \beta^7 + \beta^6 + 1)Z + 1.$$

Since the zeros of $\sigma(Z)$ are $Z = 1$ and $Z = \alpha^6$, we conclude that the error-locators are 1 and α^{17} and thus that the error-vector is

$$\mathbf{e} = (1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0).$$

We retrieve the transmitted codeword by computing $\mathbf{c} = \mathbf{y} - \mathbf{e}$.

Exercise 5.1. Do the same example with the algorithm of Berlekamp-Massey instead of Euclid's algorithm.

6. Exercises

Let C be the binary cyclic code C of length 15 with defining set $J = \{1, 3, 5\}$. In the following, $\alpha \in \mathbb{F}_{16}$ will denote a primitive element satisfying

$$\alpha^4 + \alpha + 1 = 0.$$

Exercise 6.1. Show that the complete defining set is given by

$$J(C) = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 12\},$$

and that C has generator polynomial

$$g(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}.$$

Determine the dimension of the code and apply the BCH bound on the minimum distance.

In order to find the true minimum distance of C, we will determine all codewords of weight 7.

Exercise 6.2. Write down the equations of the system $\mathcal{S}_C(7)$ and reduce the system by setting $A_0 = b$ and $A_7 = a$ and expressing everything in a, b and $\sigma_1, \sigma_2, \ldots, \sigma_7$. Compute a Gröbner basis for the ideal defined by $\mathcal{S}_C(7)$ and answer the following questions:

1. How many codewords of weight 7 does C have?
2. Determine a set M and polynomials $\sigma(X, a)$ such that $\sigma(X, a)$ has as zeros the locators of a codeword of weight 7 if and only if $a \in M$.
3. Prove that $\sigma(X, \alpha^i) = \sigma(\alpha^{13i} X, 1)$. What does this show?

We will now use code C to decode a word that is a transmitted codeword in which errors have occured. First we choose a codeword in C.

Exercise 6.3. Pick your favorite polynomial $m(X) \in \mathbb{F}_2[X]$ of degree at most 4 and encode it by computing

$$c(X) = m(X)g(X) \bmod (X^{15} + 1).$$

Now choose a random binary error-vector $\mathbf{e}$ of weight at most 3 and compute the word $\mathbf{r}$ that is received at the other end of the channel:

$$\mathbf{r} = \mathbf{c} + \mathbf{e}.$$

We will decode the received codeword using all the algorithms we have discussed. If you want you can exchange the word $\mathbf{r}$ you have chosen with someone else and try to decode the word 'he/she sent you'.

Exercise 6.4. Compute the syndromes $s_1 = r(\alpha)$, $s_3 = r(\alpha^3)$ and $s_5 = r(\alpha^5)$ and proceed with Algorithm 3.10. You have to use a computer algebra package that can compute Gröbner bases over $\mathbb{F}_{16}$. Compare your result with the codeword that was sent.

Now compute all syndromes $s_1, s_2, \ldots, s_6$ and define the syndrome polynomial

$$S(Z) = s_1 + s_2Z + s_3Z^2 + s_4Z^3 + s_5Z^4 + s_6Z^5.$$

Set

$$\sigma(Z) = 1 + \sigma_1 Z + \sigma_2 Z^2 + \sigma_3 Z^3.$$

We want to determine the σ_i such that $\sigma(Z)$ has as its zeros the reciprocals of the error positions of $\mathbf{e}$. We have seen two algorithms for this.

Exercise 6.5. Apply Sugiyama's algorithm to the situation here: compute the greatest common divisor of Z^6 and $S(Z)$ until the stop criterion of the algorithm is reached. Determine $\sigma(Z)$ from this and determine its zeros and thus the error positions. Compare your result with the codeword that was sent.

Exercise 6.6. Determine $\sigma(Z)$ by applying the Berlekamp-Massey algorithm. Again find the error-locators and compare this with your result from the previous exercise.

If the number of errors that were made during transmission is equal to 3, we can use the formulas we found by one-step decoding.

Exercise 6.7. Look up in Example 3.13 the formula corresponding to a 3-error correcting binary BCH code, substitute the syndromes you have computed, and determine the zeros and hence the error positions of the equation.

Index